AMARABAC, Journal of American Arabic Academy for Sciences and Technology, www.amarabac.com

Volume 1, Number 2, (2010), PP. 112-123

Modified background subtraction algorithm for motion detection in surveillance systems

Marwa abd el Azeem Marzouk [*]

ABSTRACT: *Recent studies have proved that a person can not watch a static scene in a monitor for more than 20 minutes therefore making traditional monitoring systems based on humans are little reliable. Most of motion detection algorithms focus on decreasing the rate of false alarms, obtaining a method for motion detection automatic and reliable. This paper presents a modified background subtraction algorithm through preprocessing frame using smoothing with RGB data format to remove noise, unwanted objects (rain and clouds) and to get more information from the frame. The preprocessed frames are used for background modeling to provide a statistical description of the entire background scene using non-recursive frame differencing technique with 4 frame differences, and scaled frame with two pre-calculated thresholds to minimize the modeling time compared to the frequent one threshold common one to give an accurate decision about moving objects in the video sequence. This modified algorithm decreased the false alarm rate by 25% to 30% and faster processing time comparing with normal background subtraction algorithm in outdoor motion detection system running 24 hours a day, through rain, sun and weak light.*

Keywords: Background subtraction، motion, frame differencing.

تعديل في خوارزم طرح الخلفية لكشف و تحديد الحركة المستخدمة في أنظمة المراقبة

مروة عبد العظيم مرزوق

الملخص: *أثبتت الدراسات الحديثة أن الشخص لا يمكن أن يراقب مشهد ثابت على شاشة لأكثر من ٢٠ دقيقة ، مما يجعل من نظم الرصد التقليدية القائمة على البشر لا يمكن الإعتماد عليها. معظم الخوارزميات المستخدمة فى الكشف عن الحركة تعمل على خفض معدل الإنذارات الكاذبة ، والحصول على وسيلة لكشف الحركة تلقائياً ويمكن الاعتماد عليها. تقدم هذه الفكرة تعديل على خوارزم طرح الخلفية المستخدم فى معظم أنظمة تحديد الحركة من خلال معالجة الإطارات القادمة من الكاميرا و إستخدامها على الصيغة RGB لإزالة الضجيج الموجود فى الصورة ، و تقليل الأجسام غير المرغوب فيها و التى قد تؤدى إلى إصدار إنذار بحدوث حركة على الرغم من عدم أهميتها بالنسب للمراقب مثل (المطر والغيوم) وللحصول على مزيد من المعلومات من الإطار ثم فى الخطوة التالية يتم استخدام الإطارات المعالجة مسبقاً لعمل نموذج لتقديم وصفا إحصائياً للمشهد بأكمله باستخدام تقنية non-recursive frame differencing مع جعل الفرق بين الإطارات التى يتم إيجاد الفرق بينها ٤ إطارات وليست إطارات متتالية للحصول على قدر أكبر من المعلومات عن طريق إيجاد هذا الفرق مع تصغير حجم الإطار لتقليل الوقت المستخدم لتحديد الأجسام التى تتحرك فى الفيديو المعروض. هذا الخوارزم المعدل أدى إلى إنخفاض معدل الإنذارات الكاذبة بنسبة ٢٥ ٪ إلى ٣٠ ٪ ووقت المعالجة أسرع مقارنة مع الخوارزم الأصلى بعد تجربة النظام بكلا الطريقتين لرصد الحركة فى مكان مفتوح و تشغيل النظام ٢٤ ساعة في اليوم ، بغرض التعرض لمختلف العوامل التى قد تؤثر على دقة النظام مثل المطر ، الشمس الشديدة ، الرياح و الاضواء الضعيفة.*

[*] Institute Of Graduate Studies and Research, Department of Information Technology, Alexandria University, Address: Sidi Bishr bahari 17 st. home No 29, E-mail: maroabdo_380@yahoo.com

INTRODUCTION

Motion detection is the simplest of the three motion related tasks, detection, estimation, and segmentation. Its goal is to identify which image points, or more generally which regions, of the image have moved between two time instants. The motion of image points is not perceived directly but rather through intensity changes. However, such intensity changes over time may be also induced by camera noise or illumination changes. Moreover, object motion itself may induce small intensity variations or even none at all.

There are many algorithms for motion detection in a continuous video stream when the camera is stationary; most of them are based on comparing of the current video frame with one from the previous frames or with something that we'll call background. This algorithm is called background subtraction. Naive description of that approach: detecting the foreground objects as the difference between the current frame and an image of the scene's static background:

$$| \text{frame}_i – \text{background}_i | > \text{Th}$$

Where the is a pre-calculated threshold value. The background image is not fixed but must adapt to:

1- Illumination changes (gradual or sudden).
2- Motion changes (camera oscillations or high-frequencies background objects).
3- Changes in the background geometry: parked cars.

One of the most common algorithms is to compare the current frame with the previous one. It's useful in video compression when it's needed to estimate changes and to write only the changes, not the whole frame. This algorithm presents an image with white pixels (motion level) on the place where the current frame is different from the previous frame. It's already possible to count these pixels, and if the amount of these pixels will be greater than a predefined alarm level (threshold) an alarm is produced about a motion event.

$$| \text{frame}_i – \text{frame}_{i-1} | > \text{Th}$$

The estimated background is just the previous frame. It evidently works only in particular conditions of objects' speed and frame rate. It is very sensitive to the threshold (Th) so that a noisy image motion will be detected in such places compared to places where there is no motion at all. If the object is moving smoothly, a small change is obtain which is less than the predefined threshold, so, it's impossible to detect moving object. Things become worse, when the object is moving very slowly, then algorithms will not give any result at all. Increasing the sensitivity (decreasing the threshold level) will produce a lot of false alarms from non moving objects like shadow, wind, small animals, etc.

Another algorithm is to compare the current frame with the first frame in the video sequence and if there were no objects in the initial frame, the comparison will detect the whole moving object independently of its motion speed. This algorithm has a lot of disadvantage for example if a stopping car seen on the first frame, but then it moved in the followed one? It'll always have motion detected on the place, where the car is.

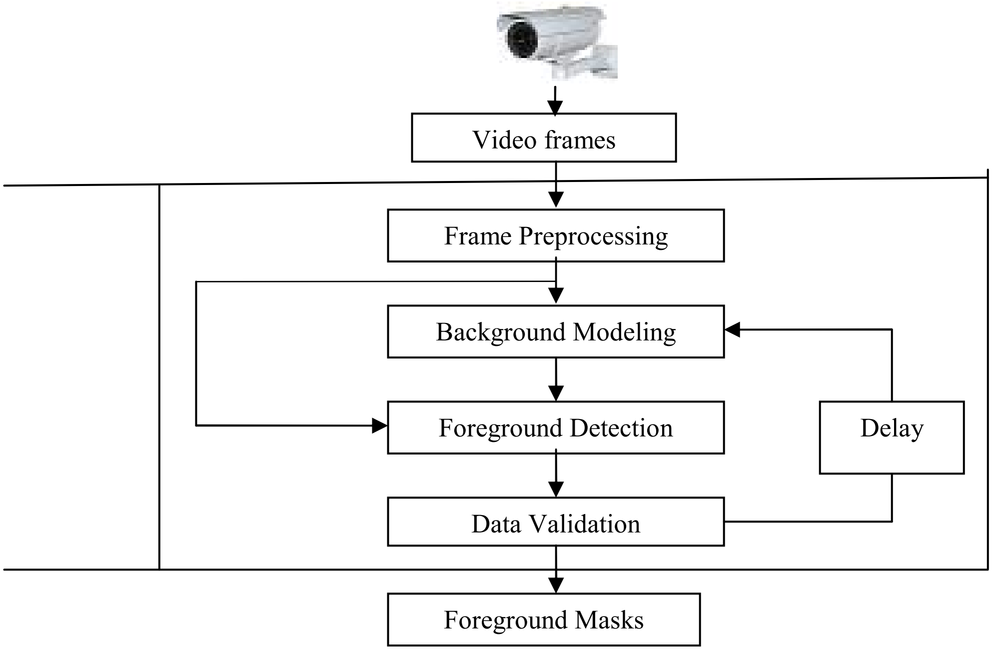
A modification for this algorithm is to renew the initial frame periodically but it still has the same problem in the cases where we can not guarantee that the renewed frame will contain only a static background.

Another method makes the background as the average or the median of the previous n frames. This method is rather fast, but very memory consuming: the memory requirement is n * size of the frame. In this background model, each pixel location is based on the pixel's recent history. In many works, such history is just the previous n frames or a weighted average where recent frames have higher weight. In essence, the background model is computed as a chronological average from the pixel's history and no spatial correlation is used between different (neighboring) pixel locations.

In the selectivity method in background subtraction algorithm at each new frame, each pixel is classified as either foreground or background. If the pixel is classified as foreground, it is ignored in the background model. This approach prevents the background model to be polluted by pixel logically not belonging to the background scene.

IMPLEMENTATION

Identifying moving objects from a video sequence is a fundamental and critical task in many computer-vision applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from a background model. There are many challenges in developing a good background subtraction algorithm. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background noise objects such as moving leaves, rain, snow, and shadows cast by moving objects. Finally, its internal background model should react quickly to changes such as starting and stopping of vehicles. Most of background subtraction algorithms follow a simple flow diagram as that shown in Figure 1.



(***Fig. 1 Flow diagram of a generic background subtraction algorithm***).

Frame preprocessing

Frame preprocessing is the first step in the Background subtraction algorithm. The purpose of this step is to prepare the modified video frames by removing noise and unwanted objects in the frame in order to increase the amount of information gained from the frame and the sensitivity of the algorithm.

Preprocessing is a process of collecting simple image processing tasks that change the raw input video into a format. This can be processed by subsequent steps. Pre-processing of the video is necessary to improve the detection of moving objects. For example, by spatial and temporal smoothing, snow can be removed from a video as shown in Figure 2. Small moving objects, such as moving leaves on a tree, can be removed by morphological processing of the frames after the identification of the moving objects, as shown in Figure 3.

Fig.2 (Video frame on left showing a traffic scene while snowing. Note the streaks in the image due to the snow flakes. The same video frame after spatial and temporal smoothing is on the right, without the snow streaks) [22].

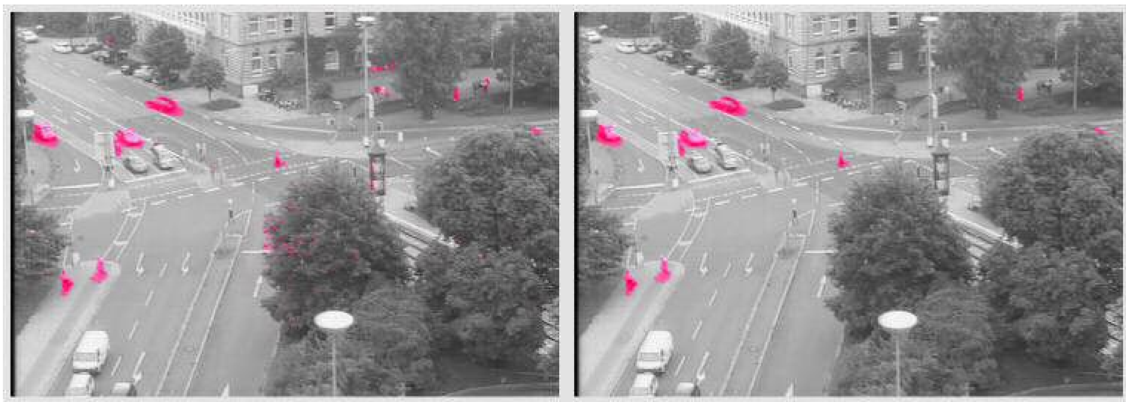


Fig. 3(The video frame on the left highlights, in pink, the objects detected as moving. Note the movement of the leaves on the trees in the foreground. Morphological processing cleans up the video frame as shown on the right) [22].

Another key issue in preprocessing is the data format used by the particular background subtraction algorithm. Most of the algorithms handle luminance intensity, which is one scalar value per each pixel. However, color image, in either RGB or HSV color space, is becoming more popular in the background subtraction algorithms.
There are six operations that can be performed:
1-Addition: As shown in Figure (I.2). This can be used to composite a new image by adding together two old ones. Usually they are not just added together since that would cause overflow and wrap around with every sum that exceeded the maximum value. Some fraction, α, is specified and the summation is performed as follows:
New-Pixel = αPixel1 + (1 − α) Pixel2

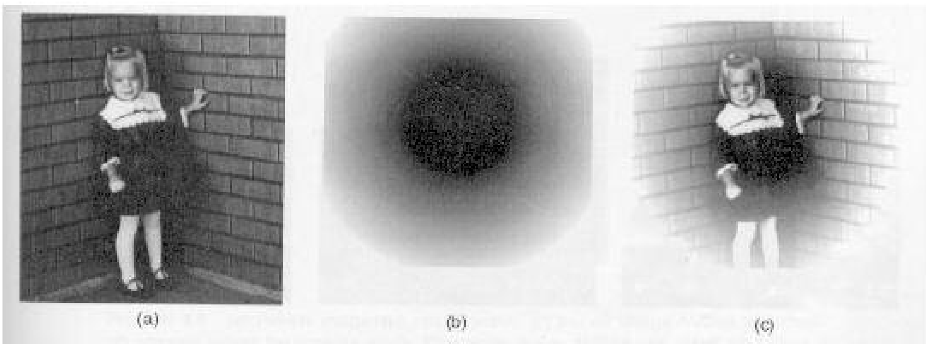


Figure (I-2) :(a) Image 1, (b) Image 2; (c) Image 1 + Image 2.

Technique that prevents overflow and allows you to be varies α so that one image can dominate the other by a certain amount. Some graphical systems have extra information stored with each pixel. This information is called the alpha channel and specifies how two images can be blended, switched, or combined in some way.

2- Subtraction: Background subtraction can be used to identify movement between two images and to remove background shading if it is present on both images. The

images should be captured as near as possible in time without any lighting conditions. If the object being removed is darker than the background, then the image with the objects is subtracted from the image without the object. If the object is lighter than the background, the opposite is done. Subtraction practically means that the gray level in each pixel in one image gray level in the corresponding pixel in the other images.
Result = x – y
where x ≥ y, however , if x < y the result is negative which, if values are held as unsigned characters (bytes), actually means a high positive value. For example:

–1 is held as 255
–2 is held as 254

A better operation for background subtraction is

$$\text{result} = |x - y|$$

i.e. x–y ignoring the sign of the result in which case it does not matter whether the object is dark or light compared to the background. This will give negative image of the object. In order to return the image to a positive, the resulting gray level has to be subtracted from the maximum gray-level, call it MAX. Combining this two gives

$$\text{new image} = \text{MAX} - |x - y|$$

3-Multi-image averaging: A series of the same scene can be used to give a better quality image by using similar operations to the windowing described in the next chapter. A simple average of all the gray levels in corresponding pixels will give a significantly enhanced picture over any one of the originals. Alternatively, if the original images contain pixels with noise, these can be filtered out and replaced with correct values from another shot.

4- Multi-image modal filtering: Modal filtering of a sequence of images can remove noise most effectively. Here the most popular valued gray-level for each corresponding pixel in a sequence of images is plotted as the pixel value in the final image. The drawback is that the whole sequence of images needs to be stored before the mode for each pixel can be found.

5- Multi-image median filtering: Median filtering is similar except that for each pixel, the grey levels in corresponding pixels in the sequence of the image are stored, and the middle one is chosen. Again the whole sequence of the images needs to be stored, and a substantial sort operation is required.

6-Multi-image averaging filtering: Recursive filtering does not require each previous image to be stored. It uses a weighted averaging technique to produce one image from a sequence of the images.

Background modeling

Background modeling and subtraction is a core component in motion analysis. The central idea behind such module is to create a probabilistic representation of the static scene that is compared with the current input to perform subtraction.

Background modeling is at the heart of any background subtraction algorithm. Much researcher has been devoted to developing a background model that is robust against environmental changes in the background, but sensitive to identify all moving objects of interest.

Background modeling uses the new video frame to calculate and update a background model. This background model provides a statistical description of the entire background scene. Background modeling techniques can be classified into two main categories non-recursive and recursive techniques.

Non-recursive Techniques

A non-recursive technique uses a sliding-window approach for background estimation. It stores a buffer of the previous L video frames, and estimates the background image based on the temporal variation of each pixel within the buffer. Non-recursive techniques are highly adaptive as they do not depend on the history beyond those frames stored in the buffer. On the other hand, the storage requirement can be significant if a large buffer is needed to cope with slow-moving traffic.

Some of the commonly-used non-recursive techniques are described below:

(A) Median filter: Median filtering is one of the most commonly-used background modeling techniques.
The background model is defined to be the median at each pixel location of all the frames in the buffer. The assumption is that the pixel stays in the background for more than half of the frames in the buffer.
The complexity of computing the median is O (L log L) for each pixel.

(B) Linear predictive filter: Linear predictive filter compute the current background estimate by applying a linear predictive filter on the pixels in the buffer. The filter coefficients are estimated at each frame time based on the sample covariance, making this technique difficult to apply in real-time.

(C) Frame differencing: Frame differencing the simplest background modeling technique, frame differencing uses the video frame at time t -1 as the background model for the frame at time t. Since it uses only a single previous frame, frame differencing may not be able to identify the interior pixels of a large, uniformly-colored moving object. This is commonly known as the aperture problem. This problem can be solved since successive frames tend to be very similar, only every fourth frame is used for statistics collection. Another simplification which increases speed is to only perform background subtraction on a sub-sampled version of each image. Thus, a 640 pixel by 480 pixel image can be rescaled to a resolu on of 160 by 120, a quarter of the linear dimensions. Since this image is one sixteenth the area of the original, the processing time for background subtraction is also one sixteenth of the original time.

Recursive Techniques

Recursive techniques do not maintain a buffer for background estimation. Instead, they recursively update a single background model based on each input frame. As a result, input frames from distant past could have an effect on the current background model. Compared with non-recursive techniques, recursive techniques require less storage, but any error in the background model can linger for a much longer period of time.

Foreground Detection

Foreground detection compares the input video frame with the background model, and identifies candidate foreground pixels from the input frame. Foreground detection then identifies pixels in the video frame that cannot be adequately explained by the background model, and outputs them as a binary candidate foreground mask.

The most commonly used approach for foreground detection is to check whether the input pixel is significantly different from the corresponding background estimate:

$$| I_t (x; y) – B_t (x; y) | > T$$

Another popular foreground detection scheme is threshold. It based on the normalized statistics:

$$\frac{| I_t (x, y) – B_t (x, y) - \mu_d |}{\partial_d} > T_s$$

Where µd and ∂d are the mean and the standard deviation of It (x, y) – Bt (x, y) for all spatial locations (x; y).
Most schemes determine the foreground threshold T or Ts experimentally.
Ideally, the threshold should be a function of the spatial location (x; y). For example, the threshold set to be smaller for regions with low contrast.
One possible modification is to use the relative difference rather than absolute difference in order to emphasize the contrast in dark areas and shadows:

$$\frac{| I_t (x, y) – B_t (x, y) |}{B_t (x, y)} > T_c$$

Another approach is to introduce spatial variability by using two thresholds. The basic idea is to identify first the “strong” foreground pixels whose absolute differences with the background estimates exceeded a large threshold. Then, foreground regions are grown from strong foreground pixels by including neighboring pixels with absolute differences larger than a smaller threshold. The region growing can be performed by using a two-pass, connected-component grouping algorithm.

Data validation

Data validation examines the candidate mask, eliminates those pixels that do not correspond to actual moving objects, and outputs the final foreground mask.

All the background models mentioned in Section 3.2 have three main difficulties:

1) They ignore any correlation between neighboring pixels.
2) The rate of background model adaptation may not match the moving speed of the foreground objects.
3) Non-stationary pixels like moving leaves or shadow cast are easily mistaken as true foreground objects.

The first problem typically results in small false-positive or false-negative regions distributed randomly across the candidate mask. The most common approach is to combine morphological filtering and connect component grouping to eliminate these regions. Applying morphological filtering on foreground masks eliminates isolated foreground pixels and merges nearby disconnected foreground regions. Many applications assume that all moving objects of interest must be larger than a certain size. Connected-component grouping is used to identify all connected foreground regions, and eliminates those that are too small to correspond to real moving objects.

The second problem has two classifications; firstly when the background model adapts at a slower rate than the foreground scene. In this case large areas of false foreground, commonly known as “ghosts", often occur. Second when the background model adapts too fast, it will fail to identify the portion of a foreground object that has corrupted the background model. A simple approach to alleviate these problems is to use multiple background models running at different adaptation rates, periodically cross-validate between different models to improve accuracy.

The third problem such as moving-leaves problem can be addressed by using sophisticated background modeling techniques like Mixture of Gaussians (MoG) and applying morphological filtering for cleanup. On the other hand, suppressing moving shadow is much more problematic, especially for luminance-only video

There are number of approximations and simplifications that have been made to increase speed of data validation. In the most important one, the timing of the algorithm has been changed so that a new background model is only generated once every 240 frames, rather than re-computing the model for each frame. During the time between model generations, statistics for the background are collected from each frame and processed incrementally. The new background model is uses on the next 240 frames. Furthermore, since successive frames tend to be very similar, therefore only every fourth frame is used for statistics collection.

Another simplification which increases speed of data validation is to perform background subtraction on a sub-sampled version of each image. Thus, a 640 pixel by 480 pixel image can be re-scaled to a resolu on of 160 by 120.This will cause one sixteenth the area of the original and then result a subtraction of one sixteenth of the original time.

EXPERMINTAL RESULT:

In this section two information retrieval measurements, recall and precision, are used to compare the modified algorithm with the original algorithm. They are defined as follows [22]:

$$Recall = \frac{\text{Number of foreground pixels correctly identified by the algorithm}}{\text{Number of foreground pixels in truth}}$$

$$Precision = \frac{\text{Number of foreground pixels correctly identified by the algorithm}}{\text{Number of foreground pixels detected by the algorithm}}$$

Recall and precision values are both within the range of 0 and 1 [20]. When applied to the entire sequence, the recall and precision reported are averages over all the measured frames. Typically, there is a trade-off between recall and precision - recall usually increases with the number of foreground pixels detected, which in turn may lead to a decrease in precision [21].
A good background algorithm should attain as high a recall value as possible without sacrificing precision.

In this paper measurements made for different environments to find the strength of each algorithm under factors which decrease the accuracy and performance of the algorithm like (fog, clouds, rain, low brightness environment and high brightness environment).
Based on these measurements, some observations regarding the background algorithms were obtained, they are as follows;

1- All background algorithms tested are sensitive to environmental noise, this is evident in the comparatively lower recall and precision numbers for the fog and rain cases.
2- Simple background subtraction algorithm adaption is faster in data validation than the modified algorithm but it gives worse results and lower accuracy with high rate of false alarms.
3- In general, algorithms that adapt more slowly often have better performance than those that adapt quickly [20][21][22].
4- The modified algorithm gives better recall and performance values than simple algorithm. This results to increased accuracy for the system by 25 to 30 .

CONCLUSIONS AND FUTURE WORK:

In this paper, some modification to the background subtraction algorithms was made. It includes preprocessing, background modeling, foreground detection, and data validation. The modified algorithm gave better accuracy with increment of up to 30 . The modifications related to environment noise, sudden change of illumination was found not encourage therefore more research is needed to improve algorithm in order to provide a balance between fast adaptation and robust modeling.

REFRENCES:

(1). A,.Mittal and D. Huttenlocher, “Scene modeling for wide area surveillance and image synthesis," in Proceedings IEEE conference on computer vision and pattern recogni on, 2, pp. 160,167, (Hilton Head Isand, SC), June 2000.

2). J. Kang, I. Cohen, and G. Medioni, \Continuous tracking within and across camera streams," in Proceedings IEEE conference on computer vision and pattern recogni on, 1, pp. 267{272, (Madison, WI), June 2003.

(3). C. Stau_er and W. Grimson, \Learning patterns of activity using real-time tracking," in IEEE Trans. on Pattern Analysis and Machine Intelligence, 22, pp. 747{57, Aug 2000.

(4). P. KaewTraKulPong and R. Bowden, \An improved adaptive background mixture model for real-time tracking with shadow detection," in Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems, Sept. 2001.

(5). M. Harville, \A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in Proceedings of the Seventh European Conference on Computer Vision, Part III, pp. 543{60, (Copenhagen, Denmark), May 2002.

(6). D. R. Magee, “Tracking multiple vehicles using foreground, background, and motion models," in Proceedings of the Statistical Methods in Video Processing Workshop, pp. 7,12, (Copenhagen, Denmark), June 2002.

(7). R. Cucchiara, M. Piccardi, and A. Prati, \Detecting moving objects, ghosts, and shadows in video streams," IEEE Transactions on Pattern Analysis and Machine Intelligence 25, pp. 1337,1342, Oct 2003.

(8). R. Pless, J. Larson, S. Siebers, and B. Westover,”Evaluation of local models of dynamic background," in Proceedings IEEE conference on computer vision and pattern recogni on, 2, pp. 73{78, (Madison, WI), June 2003.

(9). N. Oliver, B. Rosario, and A. Pentland, “A Bayesian computer vision system for modeling human interactions," IEEE Transactions on Pattern Analysis and Machine Intelligence 22, Aug 2000.

(10). I. Haritaoglu, D. Harwood, and L. Davis, \W4: real-time surveillance of people and their activities," IEEE Transac-tions on Pattern Analysis and Machine Intelligence 22, pp. 809{830, Aug 2000.

(11). B. Lo and S. Velastin, \Automatic congestion detection system for underground platforms," in Proceedings of 2001 Interna onal symposium on intelligent mul media, video, and speech processing, pp. 158{161, (Hong Kong), May 2001.

(12). Q. Zhou and J. Aggarwal, “Tracking and classifying moving objects from videos," in Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance, 2001.

(13). P. W. Power and J. A. Schoonees, “Understanding background mixture models for foreground segmentation," in Proceedings Image and Vision Computing New Zealand, pp. 267{271, (Auckland, New Zealand), Nov 2002.

(14). X. Gao, T. Boult, F. Coetzee, and V. Ramesh, “Error analysis of background adaption," in Proceedings IEEE conference on computer vision and pattern recogni on, 1, pp. 503{510, (Hilton Head Isand, SC), June 2000.

(15). D.-S. Lee, J. Hull, and B. Erol, “A Bayesian framework for gaussian mixture background modeling," in Proceedings of IEEE International Confererence on Image Processing, (Barcelona, Spain), Sept 2003.

(16). L. Fuentes and S. Velastin, "From tracking to advanced surveillance," in Proceedings of IEEE International Confer-erence on Image Processing, (Barcelona, Spain), Sept 2003.

(17). D. Gutchess et al., “A background model initialization algorithm for video surveillance," in Proceedings Eighth IEEE International Conference on Computer Vision, 1, pp. 744,740, (Vancouver, BC), July 2001.

(18). Y. Ivanov, A. Bobick, and J. Liu, “Fast lighting independent background," International Journal of Computer Vision37(2), pp. 199{207, 2000.

(19). A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, “Detecting moving shadows: algorithms and evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence 25, pp. 918 ,July 2003.

(20). A real-world system for human motion detection and tracking David Moore, California Ins tute of Technology , June 5, 2003

(21). Y. Song, A perceptual approach to human motion detection and labeling. PhD thesis, California Ins tute of Technology, 2003.

(22). Cheung, S.-C. and C. Kamath, "Robust Background Subtraction with Foreground Validation for Urban Traffic Video," EURASIP Journal on Applied Signal Processing, Volume 14, pp 1-11, 2005. UCRL-JRNL-201916.