Proceedings of Acivs 2003 (Advanced Concepts for Intelligent Vision Systems), Ghent, Belgium, September 2-5, 2003

EMSWITCH: A MULTI-HYPOTHESIS APPROACH TO EM BACKGROUND MODELLING

[1,2]*Paul J. Withagen*, [2]*Frans C.A. Groen and* [1]*Klamer Schutte*

[1]`Withagen@fel.tno.nl`
[1]TNO Physics and Electronics Laboratory, P.O. Box 96864, 2509 JG The Hague, The Netherlands
[2]IAS group, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

ABSTRACT

The detection of moving foreground objects is an important aspect of many vision applications, especially those related to video-surveillance. The Expectation Maximisation (EM) algorithm is used in many of such applications to model the background and classify object pixels.
In this paper two improvements to the EM algorithm will be given for object detection. First, it will show how to calculate background probabilities instead of binary foreground-background classifications. These probabilities will be compared to the foreground probabilities.
Second, a multi-hypothesis approach will be introduced to decide when to update the model of the background (EMswitch). This way, the model of the background is not disturbed by passing objects. At the same time it gives a accurate description of the background statistics using one or more Gaussian kernels. The standard deviation is estimated more accurately then it would be estimated using an algorithm which only updates the background model for pixels which are classified as depicting background.

1. INTRODUCTION AND RELATED WORK

A surveillance application (see for example [1, 2, 3]) usually consists of moving object detection, object tracking, and higher order processing like the detection of persons entering a prohibited area, face recognition and/or gesture recognition to determine what a person is doing. This paper concentrates on the initial step, moving object detection.
Often, background modelling is used to detect moving objects. Toyama compares a number of these algorithms in [3]. A popular algorithm is Expectation Maximisation [4]. Because of increasing speed of computers, it is now becoming feasible to make a real-time implementation [5]. The EM algorithm models the background by maintaining for each pixel a model of the appearance of colours over time. This model is described by a mixture of Gaussian kernels. For every new frame, the probability that a pixel is background is calculated by comparing its colour to this mixture model.
The advantage of an adaptive method like this is that it automatically creates a model of the background from the images and continuously updates this. This eliminates the need of initialisation with an empty background. It also guarantees that the background model is always up-to-date.
Stauffer [6] was one of the first to use the online EM algorithm for background modelling. Many authors adopted his algorithm, although some improvements were made in the update process, see for example [7].
This way of using the EM algorithm performs well for a standard case, but fails for more complicated situations like complex backgrounds and many or slow objects. In this paper two improvements are introduced to the use of the EM algorithm for the modelling of complex backgrounds. First, classification by comparing probabilities rather than labelling kernels in a mixture model is proposed. This makes classification more reliable when the foreground and background colour distributions partially overlap.
Second, EMswitch is introduced. This is an algorithm that updates the foreground and background models based on multi-hypothesis. This results in a more accurate description of the background, and consequently a more accurate foreground classification.
The outline of this paper is as follows: In section 2 the EM algorithm and its use for background modelling is shortly discussed. Two classification approaches will be described and it will be explained when to use which approach. In section 3 the EMswitch update algorithm will be introduced. This is an affordable solution for deciding when to update the background model. It is inspired by a multi-hypothesis approach. In section 4 results from the EMswitch update algorithm will be compared to other methods. Finally, conclusions will be given in section 5.

2. THE EM ALGORITHM FOR BACKGROUND MODELLING

The Expectation Maximisation (EM) algorithm is used to update a Probability Density Function (PDF) of the data.

This PDF models the occurrence frequency of colours over time for each pixel using a number of Gaussian kernels.
In subsection 2.1 a short explanation will be given of the cases in which one should use more than one kernel to describe the background. After that two algorithms will be discussed to classify between foreground and background. It then becomes clear why it is important to have an accurate model of the background statistics. The last part discusses which problems one can run into when updating the model.

2.1. The number of kernels in the mixture model

Only one kernel is needed to model a static background, but backgrounds are not always static. Examples of non-static backgrounds are:

- Water
- Moving trees and vegetation
- Computer screens and TVs

EM models such backgrounds by using more kernels. Generally, it can be said that the more complex the background is, the more kernels are needed. On the other hand, the more kernels are used, the more data is necessary for the model to converge to an accurate description of the background.
An additional reason to add kernels is when the mixture model is used not only to describe the background, but also the foreground. The model is updated for each frame, regardless whether a pixel depicts foreground or background. Using the assumption that the colour of foreground objects and the background are distinct, they will be modelled in the PDF by different kernels. Classification is done by labelling kernels background or foreground. When there are not sufficient kernels to model all occurring foreground colours, kernels describing the background will be disturbed when foreground objects pass.

2.2. Classification between foreground and background

For classification between foreground and background two algorithms will be used. The first approach is the classification algorithm proposed by Stauffer, using one model to describe both background and foreground statistics. The second approach is the algorithm we propose. It uses separate models for foreground and background statistics, allowing classification based on probabilities.

2.2.1. The Stauffer classification approach

The Stauffer classification algorithm [6] is based on the assumption that colour distributions of the foreground and background are distinct. In this way, separate kernels will be used to model the foreground and background. Classification is performed by assigning a label to each of the kernels, determining whether they model foreground or background.
First the kernels are ordered in descending order by their value of π/σ, with π the prior probability and σ the standard deviation of the kernel. The first B kernels are then chosen as the background model, where

$$B = \arg\min_b \left(\sum_{n=1}^{b} \pi_n > T_{\text{Priorsum}} \right), \tag{1}$$

with T_{Priorsum} a constant threshold. Classification is done by checking whether for colour $\vec{x}$

$$\mid \vec{x} - \vec{\mu}_n \mid < T_{\text{Stauffer}} \tag{2}$$

for any of the kernels labelled as background. T_{Stauffer} is a threshold related to the variance of the kernel, so $T_{\text{Stauffer}} = k \mid \sigma_n \mid$ with k a constant.
This classification method performs well with static backgrounds. Using it with non-static backgrounds causes problems deciding which kernel describes background when there are long occlusions. For example, if the background has value $\vec{b}_1$ in 60% of the time, and value $\vec{b}_2$ the remaining 40%, but is occluded for 30% of the time by one or more objects with a value $\vec{o}$, That would give the following prior probabilities: $\pi_{b1} = 42\%$, $\pi_{b2} = 28\%$ and $\pi_o = 30\%$. Using the prior probabilities only (This example is simplified by taking equal standard deviation for all kernels. $T_{\text{Priorsum}} = 50\%$ is used.) will result in background when either value $\vec{b}_1$ or $\vec{o}$ is measured, and foreground when value $\vec{b}_2$ is measured. This is not the desired result of the algorithm. This problem could be solved updating the background model less fast, but that would lead to problems with illumination changes.
Situations where the background and foreground colour distributions are not distinct can also cause problems. In these situations the foreground and background will not be modelled in separate kernels, so labelling of the kernels is senseless.

2.2.2. The proposed classification approach

For situations where the background is not static, or the background and foreground colour distributions are not distinct, it is proposed to compare the probabilities that the pixel belongs to the background or the foreground. Both the foreground and the background are modelled, and $P(\vec{x}|F)$ and $P(\vec{x}|B)$, the probabilities for the pixel colour $\vec{x}$ in the current frame for the hypothesis that it is foreground or background, are given by the mixture model.

These probabilities are compared and classified such that the sum of the cost for a missed detection and the cost for a false alarm is minimised. The Bayes criterion gives that a pixel should be assign to the background when:

$$\frac{P(\vec{x}|F)}{P(\vec{x}|B)} < \frac{C_B P(B)}{C_F P(F)}, \tag{3}$$

where C_F and C_B are the costs of misclassification, and $P(F)$ and $P(B)$ are the a priori foreground and background probabilities.

In this approach there is no need to label kernels as in the Stauffer approach. Classification is based on the highest probability, meaning that if the models are accurate, classification results will be optimal (i.e. the total cost of misclassification will be minimal).

To be able to use this classification approach, one model is needed for the foreground and one for the background. The background model should be per pixel, as the background will be different per pixel. Foreground objects, on the other hand, are not static. They can move over all pixels, so it makes sense to spatially share the foreground model. Only one model is used for describing all foreground objects depicted in all pixels. A model of the foreground objects can also be obtained by tracking known objects, see [1] where for each pixel the probability is calculated that it belongs to a certain object. Classification can now be performed by choosing the class (background or one of the objects) which is most probable.

2.3. Updating the model

In the previous section two separate models for the classification based on probabilities have been introduced, one to describe the foreground and one to describe the background.

The background and foreground models (in this paper the foreground model is considered static, see [1] for a model of foreground objects) need to be estimated from the data, and updated to allow for gradual changes. However, it is not trivial to decide when to update which model. A naive approximation is to update them according to the classification result, but this results in underestimating the standard deviation of the background kernels. (Assuming) to draw from a Gaussian distribution, values in the tail of the background distribution will be wrongly classified as foreground. As this does not happen frequently, it will not cause problems for the classification. However, it will cause problems when the background is not updated due to the error in the classification. The standard deviation will be underestimated and subsequently more pixels will be misclassified as foreground. This problem increases over time, see figure 1.

A second problem is that when a pixel should be foreground, but the colour is still described well enough by

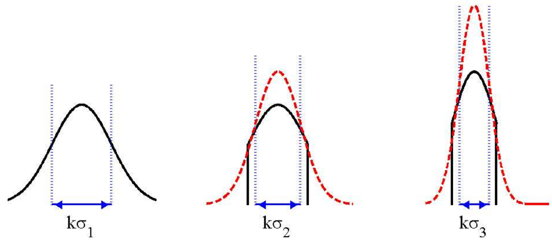


Figure 1: *When only updating a kernel when the new value is between an interval given by the mean and standard variation (dotted lines), the tail of the distribution is not updated, see the figure on the left. When only this part of the distribution is used (drawn curve in the centre figure), a new Gaussian kernel is estimated which has a lower standard deviation (dashed curve). Again, only samples within the dotted lines are used to update the kernel, so in the next frame this kernel will have an even smaller standard deviation, see the figure on the right.*

the background model to be classified as background, the background model will be updated with foreground colours, which leads to a bad representation of the real background and subsequently bad classification results. In short, updates of the background model are not wanted when a pixel shows foreground.

So on the one hand updates for values in the tail of the Gaussian distribution should not be missed, and on the other hand updates of the background model are not wanted when the pixel might depict foreground. To meet these contradicting demands, the update strategy for the models need to be considered. In the next section a multi-hypothesis approach will be discussed.

3. EMSWITCH, THE PROPOSED UPDATE ALGORITHM

To decide when and when not to update a model, time, and especially the behaviour of possible objects in time will be considered. As the goal is to classify between steady background and moving foreground, the behaviour of the foreground objects can help to decide when to update.

In the following section a multi-hypothesis approach is considered. It will be shown that it solves the problem, but at a huge cost. Then, the EMswitch update strategy, which incorporates many of the advantages of the multi-hypothesis approach with a minimum of computations and necessary memory, is introduced.

3.1. Multi-hypothesis approach

Using knowledge about possible objects, it is judged whether either the foreground or background model of a pixel in frame t should be updated. This is done using multi-hypothesis, so the decision is postponed for a number of frames to obtain more evidence for the correct classification, then the decision is made.

For updating the models it is not a big disadvantage that there is a delay, as slow changes in the background are expected. For object detection in a real-time application it would pose a problem, so the multi-hypothesis classification will not be used for this. This is not a big disadvantage, because misclassifications are caused by values in the tail of the Gaussian distributions they will not occur very frequently, and if they occur it will be mostly isolated pixels that can easily be filtered out and ignored. SO the instantaneous classification result is well enough for object detection.

It is proposed to use a multi-hypothesis approach to solve the following problems:

- When values in the tail of the distribution are not used for updates, the standard deviation will be underestimated.
- When an object stays at a certain location, it should be regarded as changed background instead of an object.

This induces demands on the period that objects are visible. An object should be visible in a certain pixel during N_F frames, where

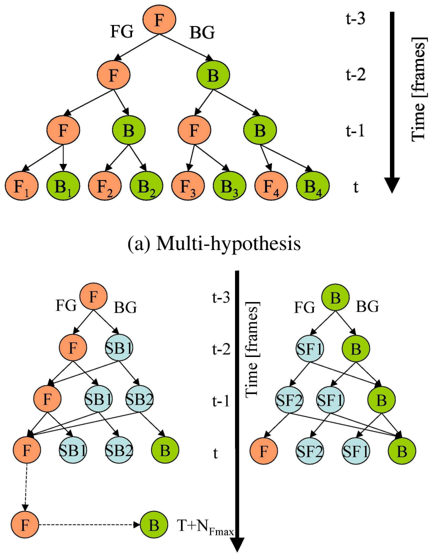
$$N_{Fmin} < N_F < N_{Fmax}, \tag{4}$$

with N_{Fmin} small, a few frames and N_{Fmax} large, about thousand frames.

Objects which are depicted shorter than N_{Fmin} frames by a pixel will not be detected, while when a pixel depicts foreground longer than N_{Fmax} frames it will be considered changed background. When this happens, a model of this new background should be available. Between each object (or group of objects) it is expected to see the background for at least N_{Bmin} frames, otherwise it is considered to be one object and the background is not updated in between the sub-objects. N_{Bmin} can be small, a few frames. As N_{Fmin} and N_{Bmin} are small and N_{Fmax} is large, these assumptions are not too tight for image sequences in surveillance applications.

A multi-hypothesis tree is now built (see figure 2(a)). Instead of just updating the model for each frame according to classification, all hypotheses are split, whereby the foreground model is updated in the one, and the background model is updated in the other. After a number of frames N_{MH} (at least the maximum of N_{Bmin} and N_{Fmin}, but preferably as much as N_{Fmax}), the most probable branch from the multi-hypothesis tree is selected by using the knowledge about the life-time of the objects given above. The other branch is removed (pruned).

This solves both problems, but at a huge cost. The complexity of this algorithm is $O(2^n)$ with n in the order of the maximum of N_{Fmin} and N_{Bmin}.



(a) Multi-hypothesis

(b) EMswitch

Figure 2: *This figure shows the possible state transitions using multi-hypothesis with a memory of three frames and EMswitch for which both N_{Fmin} and N_{Bmin} are equal to two, so after three sequential classifications other than the initial state, the state is changed from* **B** *to* **F** *or vice versa. The number gives the number of frames the pixel has been in a switch state, so* **SF**2 *means that the pixel has been in switch state* **SF** *for two frames. The states* $\mathbf{B}_3$, $\mathbf{F}_2$, $\mathbf{F}_3$ *and* $\mathbf{F}_4$ *from multi-hypothesis cannot be used in EMswitch because only two background models are kept in memory simultaneously, one for the stable state* **B** *and one for the switch states* **SF** *or* **SB**.

3.2. The EMswitch update strategy

In this section the EMswitch update strategy is introduced. It is derived from the multi-hypothesis approach described above, but only two simultaneous background models and one foreground model will be used.

Looking at the differences between the foreground and background model, it can be see that as the foreground model is shared by all pixels, it poses no problem when it is not always updated. This model can be updated when it is certain that the pixel depicts foreground. When there is doubt the model will not be updated. The background model on the other hand, should be updated as often as possible. In case of doubt a copy of the current model is made before updating. When it shows that the update was wrongly performed, the original model is restored by means of the copy that was made.

EMswitch uses four states: **B** (background), **F** (foreground), **SB** (switch to background) and **SF** (switch to foreground). As long as the state of a pixel is **B** or **F** and the classification result is the same, the state remains and the corresponding model is updated. In these cases there is no restoration possible. When the classification result is different from the state of a pixel, the state of that pixel is

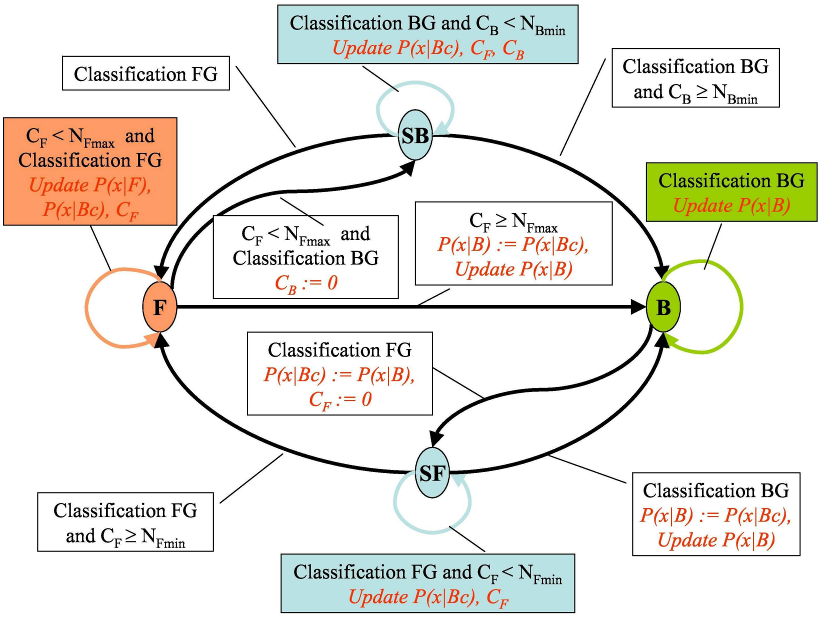


Figure 3: *This figure shows the possible transitions in EMswitch between each state. Conditions are printed in black, actions in red italics.*

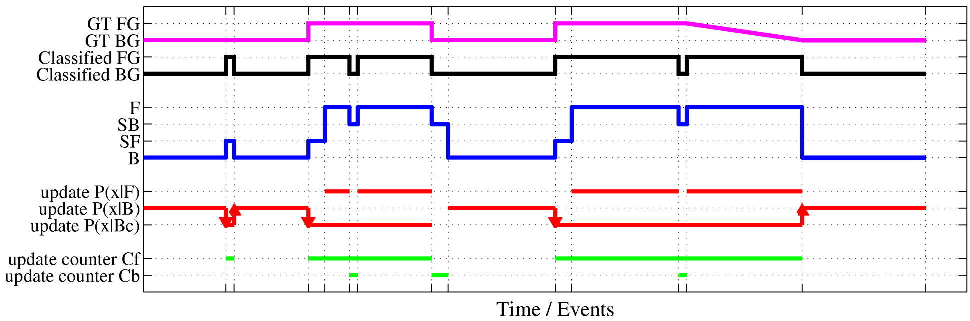


Figure 4: *This figure shows the state transitions in EMswitch for one pixel in a synthetic sequence. Five graphs are shown, from top to bottom: ground truth classification, EM classification, current state of EMswitch, models which are updated in EMswitch, and counters which are updated in EMswitch. Arrows indicate the copy or restore of a model.*

changed to an intermediate state (from **B** to **SF** and from **F** to **SB**) indicating that there is a possible switch from **F** to **B** or vice versa (see also figures 3 and 4).

When the state of a pixel was **B** and the pixel is classified as foreground a copy of the background model is made, the foreground counter C_F is reset and the state is changed to **SF**. In this state counter C_F and the copy of the background model $P(\vec{x}|Bc)$ are updated for each frame. The state of the pixel is not changed until either the counter C_F reaches N_{Fmin}, in which case the new state becomes **F**, or when a pixel is classified as background, the state transfers back to **B** and the model of the background is replaced with the copy.

For a pixel of which the state transfers to **SB** the counter C_B is reset. In state **SB** counter C_B is updated for each frame. The state of the pixel remains **SB** until the counter C_B reaches N_{Bmin}, in which case the state transfers to **B**, or until the pixel is classified as foreground. The state then returns to **F**.

The rules mentioned above solve the problem of values in the tail of a distribution. To cope with a possible change of background counter C_F and the copy of the background model $P(\vec{x}|Bc)$ are updated in states **F** and **SB**. When the counter C_F reaches N_{Fmax} the background is considered changed. The background model is written over with the updated copy and the state of the pixels transfers to **B**. In this way, as soon as a change of background is assumed, the new background model is available.

3.2.1. Comparison to multi-hypothesis

As can be seen in figure 2(b), the algorithm introduced can be seen as a partial multi-hypothesis algorithm. When the state of a pixel is **SB**2 the state can transfer to **F**, but it is not possible to go one step back to **SB**1. The same holds for state **SF**. A problem that remains with EMswitch occurs when a pixel depicts foreground, and while its state is still **SF**, the pixel is erroneous classified as background for one frame. The foreground colour will be updated in the background model. This will however only occur when the object closely resembles the background.

3.2.2. EMswitch initialisation

The mixture model describing only the background has to be initialised, but how can it be decided whether or not a pixel belongs to the background if there is no background model? This can be solved by starting with Stauffer classification, and use this classification to initialise the background model.
A state **I** (Initialisation) and a counter C_I are introduced. All pixels start with state **I**. The initialisation is as follows:

- Use the first *frames* (until $C_I > N_{I1}$) only to update an additional model $P(\vec{x})$.
- Until $C_I > N_{I2}$, i.e. the next number of *updates*, use $P(\vec{x})$ to classify between foreground and background using Stauffer classification. Update C_I and $P(\vec{x}|B)$ only for frames for which the classification result of the pixel is background and update $P(\vec{x})$ for all frames.
- Now set the highest prior probability to 0.99 and all others to $\frac{0.01}{nKer-1}$, with $nKer$ the number of kernels in the mixture model.
- Start using the combination of $P(\vec{x}|F)$ and $P(\vec{x}|B)$ for classification. End of initialisation.

In the initialisation phase the Stauffer classification is used as it can handle a mixed model of foreground and background.

4. EXPERIMENTAL RESULTS

In this section the results on some real-life test sequences will be shown. The results of three methods are compared:

- *AlwaysUpdate*, always update the model, classify using Stauffer classification.
- *UpdateBackgroundOnly*, update the model only when the classification result is background, classify using Stauffer classification.
- *EMswitch*, update using the EMswitch algorithm, classify using probabilities.

Updating the models is always done by using the original online EM algorithm, since there are some problems in the way Stauffer updates, see for example [7].
The classification results for several frames of the test sequences and numerical results to evaluate which method performs best will be given. First, the image sequences used will be described and some implementation details and settings of the algorithm will be given. Subsequently, the classification results will be shown and discussed.

4.1. Image sequences and ground truth labelling

Two real-life image sequences were used:

- *Intratuin*, Parking lot with moving tree, 1300 frames.
- *PETS 2001*, DATASET 3, TRAINING, CAMERA1. Dataset from the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS). The images contain small and large objects, and there are large changes in illumination intensity. 5563 frames.

Both sequences are colour video data with eight bit per colour. They were processed with the same parameter settings, without additional tuning.
A number of frames were manually labelled: For PETS2001 every 500th frame. For the Intratuin sequence every 250th frame was labelled. Each pixel was labelled as one of: foreground, background, or any. Classified frames were compared to the corresponding labelled image, and the ratios of correctly classified foreground over total foreground pixels (R_F) and correctly classified background over total background pixels (R_B) were determined. For the EMswitch algorithm this was additionally performed for the stable states (**F** and **B**) and switch states (**SF** and **SB**) separately, because the stable states were expected to perform much better than the switch states. By using these ratios, an objective evaluation can be performed between different parameter settings and between different algorithms.

4.2. Implementation details

In all experiments the following parameters settings were used as defaults:

- RGB colour space and 3D Gaussian kernels with diagonal covariance matrices were used.
- $P(\vec{x}|F)$ was taken to be uniformly distributed with a value: $P(\vec{x}|F) = \frac{1}{256^3}$.
- Number of kernels: nKer= 5.
- Thresholds: $T_{\text{priorsum}} = 0.5$, $T_{\text{Stauffer}} = \sqrt{5} \mid \sigma \mid$.
- N's: $N_{I1} = 100$, $N_{I2} = 200$, $N_{Fmax} = 350$, $N_{Fmin} = 3$, $N_{Bmin} = 3$.

- Update speed of the EM algorithm $= 0.01$ (the higher this value, the faster the model reacts on the data).
- $P(B) = 1 - P(F) = 0.5$ and $\frac{C_B}{C_F} = 0.05$.
- For isolated pixels the classification result was changed to that of the surrounding pixels.

4.3. Classification results

The results for the AlwaysUpdate and the EMswitch algorithms on the test sequences are shown in figure 5. It can be seen that both algorithms perform comparably well on the “PETS2001” sequence. On the “Intratuin” sequence the model updated using EMswitch performs better. When using AlwaysUpdate there are large holes in the objects caused by earlier passing cars. As the model was updated when the cars passed by, the background model was disturbed.

In figure 7 ROC curves are shown for different settings. The update speed was varied between 0.0001 and 0.1, and the detection thresholds between 1 and $\sqrt{15}$ from their initial settings. It can be seen that for UpdateBackgroundOnly (which, as explained in section 2.3, has problems) performs better than expected for the “Intratuin” sequence, although it is expected that it does not model the statistics of the background very accurately. Figures 7(c) and 7(d) show the sensitivity to variations in the update speed. The closer the points are together, the less sensitive the algorithm is. As these figures illustrate, the EMswitch algorithm is more robust for parameter settings than both other methods.

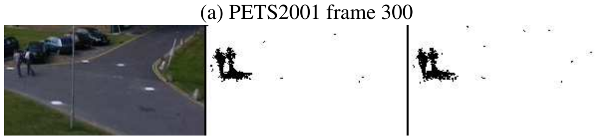
The robustness of EMswitch for the update speed is caused by foreground objects not being updated in the background model, so their colours will not become part of this model. Only when objects are at a certain location for such a long time that it can be said that the background must have changed (longer than N_{Fmax} frames), the background model is renewed.

Because the background model is not updated for foreground objects, the update speed can be chosen to be relatively high, which is an advantage when there are fast illumination changes. In figure 6 the results are shown of processing the “Intratuin” sequence with a ten times higher update speed. As can be seen, most objects are lost when using AlwaysUpdate.

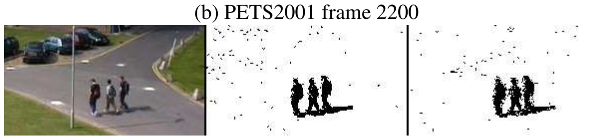
The robustness of an algorithm can also be tested by using optimal parameter settings of one image sequence on another, and comparing the results to those using optimal settings for this image sequence. As shown in table 1, EMswitch has less than one fourth the performance decrease as both other update algorithms.

In figure 7(e) ROC curves are drawn for EMswitch and the stable and switch states. It is clear that the majority of the errors are in the switch states. For applications where

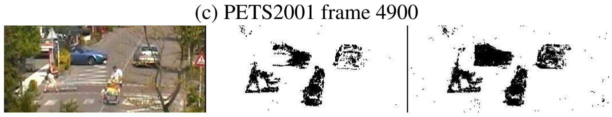
(a) PETS2001 frame 300



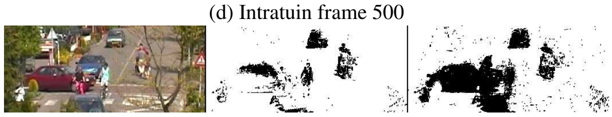
(b) PETS2001 frame 2200



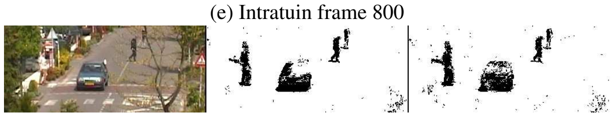
(c) PETS2001 frame 4900



(d) Intratuin frame 500



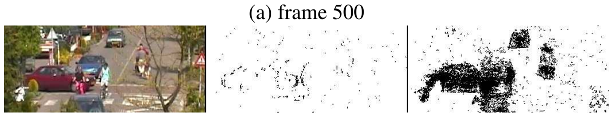
(e) Intratuin frame 800



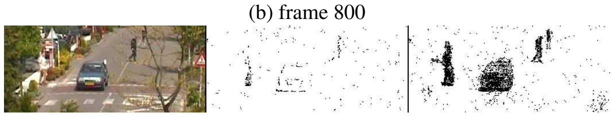
(f) Intratuin frame 1100

Figure 5: *Classification results. For each frame the input frame, the results of AlwaysUpdate using Stauffer classification, and the results of using EMswitch updating with probability classification are shown respectively. White pixels depict background, black pixels foreground.*

(a) frame 500



(b) frame 800

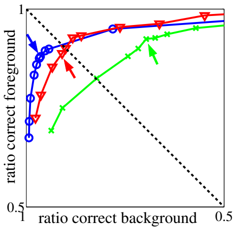


(c) frame 1100

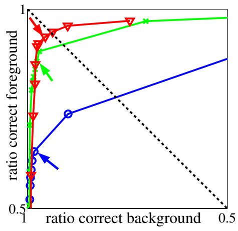
Figure 6: *Results on the “Intratuin“ sequence with a ten times higher update speed.*

Table 1: *Different image sequences may have different optimal parameter settings. If there is a large difference in performance between using the optimal settings for one sequence and using the optimal setting for another, the method is very sensitive to the parameter setting. The numbers in this tables are the average of R_F and R_B.*

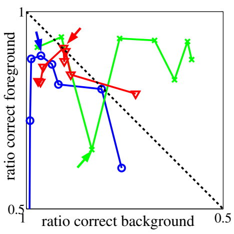
Image sequence:	PETS2001 sequence			Intratuin sequence		
Parameter setting optimal for:	PETS2001	Intratuin	Difference	PETS2001	Intratuin	Difference
AlwaysUpdate	93.4 %	80.6 %	12.8 %	61.5 %	83.7 %	22.2 %
UpdateBackgroundOnly	94.1 %	76.2 %	17.9 %	78.0 %	93.3 %	15.3 %
EMswitch	90.7 %	89.7 %	1.0 %	91.5 %	94.4 %	2.9 %



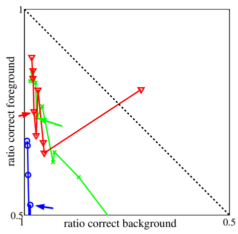
(a) PETS2001, threshold



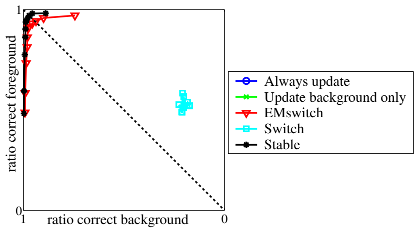
(b) Intratuin, threshold



(c) PETS2001, update speed



(d) Intratuin, update speed



(e) Intratuin, thresholds, EMswitch states

Figure 7: *ROC curves for variation of the update speed and detection thresholds. The closer to top-left, the better (note the axes limits). Arrows indicate the default settings. When the cost of falsely detected and missed object pixels are equal, lines of equal performance are perpendicular to the black dashed line.*

there can be a small time shift, for example with non real-time applications, classification of EMswitch can be improved by using this information.

5. CONCLUSIONS

In this paper a novel classification method for object detection has been introduced using Expectation Maximisation. This method uses separate models for the background and foreground. Classification is done by comparing background and foreground probabilities.
It was demonstrated that to obtain separate models for background and foreground, a form of reasoning is necessary to update the background model, because the standard deviation is otherwise not estimated correctly. To update the background model EMswitch has been introduced: an update strategy based on multi-hypothesis using life-time knowledge about the expected objects.
Numerical results were provided for two real-life image sequences that show that probability-based classification performs equally well or better than Stauffer classification which uses a shared model for foreground and background. It was also demonstrated that the method introduced is less sensitive to the setting of the update speed and detection threshold.

6. REFERENCES

[1] Paul J. Withagen, Klamer Schutte, and Frans C.A. Groen, "Likelihood-based object tracking using color histograms and EM," in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2002)*, A.M. Tekalp, Ed., Rochester, NY, Sept. 2002, pp. 589–592, IEEE.

[2] D. Harwood I. Haritaoglu and L.S. Davis, "W^4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analisys and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, August 2000.

[3] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *IEEE Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 255–261.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM-algorithm," *Journal of the Royal Statistical Society*, vol. B 39, no. 1, pp. 1–38, 1977.

[5] C. Priebe, "Adaptive mixtures," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 796–806, 1994.

[6] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, 1999, vol. 2, pp. 246–252.

[7] I. Pavlidis, V. Morellas, P. Tsiamyrtzia, and Steven A. Harp, "Urban surveillance systems: from the laboratory to the commercial world," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1478–1497, Oct. 2001.