Cascaded change detection for foreground segmentation

Luís F. Teixeira and Luís Corte-Real
FEUP / INESC Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
{luis.f.teixeira,lreal}@inescporto.pt

Abstract

The extraction of relevant objects (foreground) from a background is an important first step in many applications. We propose a technique that tackles this problem using a cascade of change detection tests, including noise-induced, illumination variation and structural changes. An objective comparison of pixel-wise modelling methods is first presented. Given its best relation performance/complexity, the mixture of Gaussians was chosen to be used in the proposed method to detect structural changes. Experimental results show that the cascade technique consistently outperforms the commonly used mixture of Gaussians, without additional post-processing and without the expense of processing overheads.

1. Introduction

The extraction of moving objects from a visual sequence is a very important operation in many vision systems. Typical applications include real-time analysis of visual scenes in order to identify events and actions, such as visual surveillance and human-machine interface systems. Moreover, the extraction of video objects is also useful for video editing applications, among others. Probably due to its simplicity, the most common approach for discriminating a moving object from the background is *background subtraction*. The rationale is the subtraction of the current image from a reference image, which is somehow acquired in a step prior to subtraction. Non-changing segments of the image are then considered as being part of the background, whereas the foreground consists of the changing segments, including moving and new objects. However, if the reference is not modelled or updated adequately this technique can be highly susceptible to environment conditions like illumination changes. For example, a straightforward way of acquiring a reference image would be by obtaining a statistical representation of the previous N frames (e.g. pixel-wise average image). In fact the techniques most frequently employed rely on the sequence's previous "history" to obtain a suitable *model*. After having the reference image, the segmentation would be completed by a simple thresholded subtraction operation. This naive approach, despite its processing efficiency, may not be adequate for real-world systems, or at least for non-controlled environments. Changes in illumination conditions and dynamic behaviour in the background may result in unacceptable rates of false positives. More complex techniques are therefore needed, in order to achieve robust *background modelling*. Nevertheless, it is important to stress that this operation is often required to perform as fast as possible, since it is usually the first step in a processing chain, assembled to acquire higher level semantic knowledge. Overly complex modelling schemes may reveal themselves unfeasible despite performing at low error rates.

The different approaches to background subtraction differ in the way the reference background is modelled and how the model is updated. Ideally, the performance should not depend on the camera placement, nor should it be sensible to what happens in its visual field or to lighting effects. It should be capable of dealing with movement through cluttered areas, objects overlapping in the visual field, shadows, lighting changes, effects of moving objects in the scene, slow-moving objects and objects being introduced or removed from the scene.

Existing methods for background modelling may be classified as predictive or non-predictive. Predictive methods model the scene as a time series and develop a dynamic model to recover the current input based on past observations. Kalman filters [6][13] are usually employed to update slow and gradual changes in the background, thus these methods are mainly applicable to backgrounds consisting of stationary objects.

Non-predictive methods for background modelling do not consider the order of input observations and build a probabilistic representation (p.d.f.) of the observations at a particular pixel. Methodologies of this type include the use of a unimodal distribution (usually a Gaussian)[14]. If each pixel resulted from a particular surface under particu-

IEEE Workshop on Motion and Video Computing (WMVC'07)
0-7695-2793-0/07 $20.00 © 2007 IEEE

lar lighting, a single Gaussian would be sufficient to model the pixel value accounting for acquisition noise. Moreover, if only lighting changed over time, a single, adaptive Gaussian model per pixel would be sufficient. In practice this does not happen and sometimes a more complex modelling is needed, which is the case of mixture of Gaussians (MoG) [12][7]. In [3], a non-parametric model is proposed, where a kernel-based function is used to represent each pixel's colour distribution. The kernel-based distribution is a generalization of MoG which does not require parameter estimation. In [5], a similar approach is followed, where the distribution of temporal variations in colour at each pixel is used to model the background.

More recent approaches to background modelling include the principal features[8] approximation that considers only the more relevant features to create a model and the mean-shift method[4][11], which was also previously applied to image segmentation.

Other techniques combine temporal and spatial modelling. In [9], a mixture model (Gaussians or Laplacians) is used to represent the distributions of background differences for static background points. A Markov random field (MRF) model incorporates the spatial coherence for robust foreground segmentation.

2. Cascaded foreground segmentation

Before detailing the proposed algorithm let us first consider the following: when no structural change occurs, the difference between a pixel value, represented by a colour vector $\boldsymbol{v}$, in the current frame $\mathcal{F}^c$ and a reference frame $\mathcal{F}^r$ can essentially result from two factors – illumination variation or noise. Illumination variation can be accounted by a positive multiplicative factor k which modulates the signal, while noise can be accounted by a superimposed vector $\boldsymbol{v}_N$, modelled by a Gaussian or Laplacian distribution. – Figure 1 represents this in a two-dimensional space.

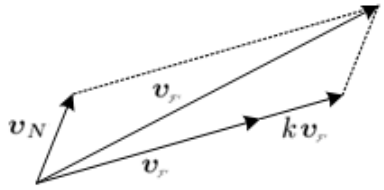

Figure 1. Effects of illumination variation and noise over a reference colour vector.

Considering that the sophomore cause can be successfully eliminated, we are left with the first. Hence, when a pixel change results solely from illumination variation, its colour vector is necessarily collinear with the reference colour vector and a simple test can be used to identify illumination variation-induced changes. Therefore, we will first address how we can effectively remove typical noise introduced by the capture process in order to guarantee colour vector collinearity.

2.1. Identification of noise-induced changes

Assuming that we know the reference frame $\mathcal{F}^r$, we will first address how we can effectively remove typical noise introduced by the capture process. For that purpose we use a method proposed by Aach [1] which states that it is possible to assess what is the probability that a value at a given position, in a given image, is due to noise instead of other causes when compared to another image. It is assumed that the additive noise affecting each image results from a Gaussian process with mean μ_N and standard deviation σ_N. Also, noise affecting successive images in the sequence is considered as uncorrelated. The standard deviation σ_N can be obtained by computing the statistics of the difference $d_{(i,j)}$ for each pixel (i,j) between the reference image and the current image. Now, consider a window W^n containing n pixels around the pixel under evaluation with $\Delta^2_{(i,j)} = \sum_{(k,l)\in W^n_{i,j}} d^2_{(k,l)}$. It can be shown that the corresponding random variable Δ^2 follows a χ^2 distribution. Given the hypothesis H_0 that $\Delta^2_{(i,j)}$ results from noise and not from other factor, the probability that hypothesis H_0 is satisfied is given by:

$$P(\Delta^2 > \Delta^2_{(i,j)}|H_0) = \frac{\Gamma(\frac{n}{2}, \frac{\Delta^2_{(i,j)}}{2\sigma_c^2})}{\Gamma(\frac{n}{2})} \qquad (1)$$

with $\sigma_c^2 = 2\sigma_N^2$ and where $\Gamma(n/2)$ is the Gamma function. The choice for the window size n must have into consideration the trade-off between noise sensitivity and foreground edge definition. Nevertheless, all experiments were performed using a window size of $n = 25$. When the estimated probability in equation (1) is smaller than a threshold T_N we consider that H_0 is not satisfied at the pixel position (i,j).

Whereas for pixels that validate the hypothesis H_0 we guarantee that changes were originated solely by camera noise, for others we can safely assume that the any effect of noise is negligible when compared to any other change. In other words, if a pixel's colour vector is being modified by illumination variation and no structural change, we have $\boldsymbol{v}_{\mathcal{F}^c} \simeq k\boldsymbol{v}_{\mathcal{F}^r}$.

This test defines a first set of pixels that can potentially be part of the foreground because all pixels that satisfy H_0 are necessarily part of the background and are marked as such for the current frame; all others need further analysis.

2.2. Identification of illumination variation-induced changes

After discarding noise-induced changes, a simple collinearity test is performed. As previously stated, with this test any modification introduced by illumination variation is discarded. The test consists in evaluating the angle between the current pixel colour vector $\boldsymbol{v}^c$ and the reference colour vector $\boldsymbol{v}^r$.

$$\cos\theta = \frac{\boldsymbol{v}^c \cdot \boldsymbol{v}^r}{\|\boldsymbol{v}^c\| \, \|\boldsymbol{v}^r\|} \tag{2}$$

If $\cos\theta$ is smaller than a threshold T_I very close to 1, than the vectors are not considered to be collinear and the test is not validated. In practice, this test can become unstable and to overcome this problem we consider a second threshold in the noise-identification test; the second threshold usually is much smaller than T_N. The identification of illumination variation-induced changes is therefore only applied only to pixels that fall between both these thresholds.

The set of potential foreground pixels is refined by marking as background the pixels that validate this second test. Pixels that have a probability less than the second threshold in the previous test are maintained as foreground.

2.3. Identification of dynamic background behaviour

Finally, the probability estimation that a pixel belongs to the background (4) can be performed only in the pixels that do not validate the statistical (1) nor the collinearity (2) tests, resulting in a considerable reduction in execution time of the algorithm. This is especially true for typical surveillance streams where the relevant moving objects occupy a small fraction of the entire field of view. Another positive effect of the proposed modification is the drastic reduction of small artefacts which often need to be removed in common mixture of Gaussians modelling by morphological filtering (e.g. connected operators).

To model and identify the dynamic background behaviour we will be using pixel-wise background estimation approaches, as presented in the introduction section. Several techniques can be used for this purpose and we will be considering and testing five of them.

1) Running Average (RAvg) – The background is modelled as the average of the previous frames but, in order to avoid expensive memory requirements, this average is approximated by an adaptive filter with a learning rate α. Each background pixel value at position (i, j) and time instant t is given by:

$$B_{(i,j)}(t) = \alpha I_{(i,j)}(t) + (1-\alpha) B_{(i,j)}(t-1)$$

Foreground is then estimated using a thresholded subtraction of the current frame and the estimated background. This technique is probably the most naive but has a very simple and very fast implementation. The results are therefore far from good in particular with complex backgrounds. Since we are considering only a static representation of the background to perform the subtraction, whenever some kind of dynamic behaviour in the background happens it will be incorrectly classified as foreground. Nevertheless, the running average should represent the minimum acceptable performance for these types of algorithms. All tests were performed with a threshold T_{RAvg} of 15.

2) Mixture of Gaussians (MoG) – Instead of estimating the background representation directly, another and more effective approach is to estimate a background model that can predict the behaviour in each pixel, using the pixel's "history". By estimating the background probability density function (p.d.f.) we are able to do just that. Assuming that any structural changes affecting the value of the pixel are caused by several processes, each modelled by a Gaussian, we can therefore define the probability of observing its value as:

$$P(\upsilon_t) = \sum_{k=1}^{K} P(G_k) P(\upsilon_t | G_k) = \sum_{k=1}^{K} \omega_k \cdot \eta(\upsilon_t, \mu_k, \sigma_k) \tag{3}$$

where G_k is the k-th Gaussian of K distributions, ω_k, μ_k and σ_k are, respectively, an estimate of the weight, the mean value and the variance of the k-th Gaussian in the mixture; η is the normal density function. Moreover, it can be easily shown [7] that, given the current colour vector υ_t in a pixel, the probability that the pixel belongs to the background is:

$$P(B|\upsilon_t) = \frac{\sum_{k=1}^{K} P(\upsilon_t|G_k) P(G_k) P(B|G_k)}{\sum_{k=1}^{K} P(\upsilon_t|G_k) P(G_k)} \tag{4}$$

If $P(B|\upsilon_t) > T_{MoG}$ the pixels that validate this test are also marked as background and the remaining pixels form the definitive set of foreground pixels for the current frame. However, two density estimation problems are left to resolve: firstly, estimating the distribution of all observations, within a period of time, at each pixel location using a Gaussian mixture (3), which provides estimates of both $P(G_k)$ and $P(\upsilon_t|G_k)$; and secondly, evaluating how likely each Gaussian in the mixture represents the background, i.e., $P(B|G_k)$. To accomplish the first estimation, In [12] an online K-means approximation is proposed in order to model pixel variation over time by a mixture of Gaussians, as given by Equation (4). It uses a fixed learning rate to update each Gaussian's parameters over time and a Gaussian substitution algorithm whenever no match is possible. However, using a fixed learning rate can often result in slow

convergence. Following the same rationale, and in order to improve the convergence speed, in [7] it is proposed an adaptive learning rate schedule for each Gaussian. The estimation of $P(B|G_k)$ is based on application-specific heuristics; we will be using this last approach.

The background image representation can be defined as the expected value of the background process. Thus, the background pixel at (i, j) and time t is defined by $E[\upsilon_{i,j,t}|B]$ which is evaluated by a weighted average of the Gaussian means.

$$E[\upsilon_{i,j,t}|B] = \frac{\sum_{k=1}^{K} \mu_k P(B|G_k)P(G_k)}{\sum_{k=1}^{K} P(B|G_k)P(G_k)} \quad (5)$$

The tests with MoG were done with the following parameters: $K = 3$, $\alpha = 0.005$ and $T_{MoG} = 0.05$.

3) Kernel Density Estimation (KDE) – It is possible to approximate each background's pixel p.d.f. by the histogram of the most recent values classified as background. This approach has however some problems. Namely, being the histogram a step function, the p.d.f. modelling can reveal itself erroneous. In [3], it is proposed a non-parametric model based on Kernel Density Estimation (KDE). KDE guarantees a smoothed, continuous representation of the histogram. The background p.d.f. is given as a sum of Gaussian kernels centred in the most recent N background values, xi:

$$P(\upsilon_t) = \frac{1}{N} \sum_{k=1}^{N} \eta(\upsilon_t - \upsilon_k, \Sigma_k) \quad (6)$$

Even if background values are not known, unclassified sample data can be used instead. This inaccuracy will be recovered along model updates. Given 6, the pixel with the colour vector υ_t is classified as foreground if $P(\upsilon_t) < T_{KDE}$, where T is a global threshold. An important issue in KDE is the estimation of Σ_k - the kernel bandwidth. In [3], it is considered a diagonal matrix for simplicity and each variance is estimated in the time domain by analysing the set of differences between two consecutive values

Model update consists in selectively updating the vector of the previous N background values. The model proposed in [3] also considers the use of two concurrent similar models, one for long-term and the other for short-term memory. In addition, spatial correlation is taken into consideration by the model. However, we will not be considering both these modifications since we are comparing pixel-wise modelling techniques. These types of considerations should be transversal to all algorithms we are evaluating. The tests executed with this algorithm used the following parameters: $N = 50$ and $T_{KDE} = 10^{-6}$.

4) Principal Features (PF) – More recently, other approaches were proposed to estimate the background p.d.f.. In [8], the background is represented at each pixel by the most frequent features, or *principal features*.

The classification is done using a Bayesian framework, and it is shown that a pixel represented by $\boldsymbol{\upsilon}$ is classified as belonging to the background if:

$$2P(\boldsymbol{\upsilon}|B)P(B) > P(\boldsymbol{\upsilon}) \quad (7)$$

Otherwise, it is classified as belonging to the foreground. We need however to know *a priori* or estimate the probabilities $P(\boldsymbol{\upsilon}|B)$, $P(B)$ and $P(\boldsymbol{\upsilon})$. As stated previously, one way to estimate these probabilities is to use a histogram of features. The important contribution of [8] is that it proposes that these probabilities can be estimated using solely the most representative features in the histogram, given that these can represent the background effectively. Therefore, for a proper selection of features, there would be a small value N of features (the principal features) that can approximate well the background by $\sum_{k=1}^{N} P(\upsilon_k|B)$.

The learning and update process is done using a table of statistics for the possible principal features of the background. The update of estimated probabilities through time is done using a simple adaptive filter according to the type of change that occurred (gradual or "once-off").

Note also that the algorithm proposed in [8] uses several types of features, namely: spectral, spatial and temporal features. For this comparison we will be using only the spectral features, i.e. colour information. Otherwise, the results for this algorithm would be biased. The tests executed with principal features used the following parameters: $\alpha = \beta = 0.04$ (rate for probability and background learning, respectively), $M = 50$, $N = 20$ and $M1 = 0.75$ (for "once-off" detection).

5) Mean Shift (MS) – The mean shift technique is an iterative gradient-ascent method that allows it to detect modes of a multimodal distribution and their covariance matrix. The only parameter needed is the bandwidth range that is application-specific. The mean shift algorithm states that, for a given set of points $x_i, i = 1, \ldots, n$, the mean shift vector in the one-dimensional case can be expressed as:

$$m(x) = \frac{\sum_{i=1}^{n} x_i g\left(\frac{x-x_i}{h}\right)^2}{\sum_{i=1}^{n} g\left(\frac{x-x_i}{h}\right)^2} - x$$

where x is an arbitrary pont in the data space, h is a positive value called the analysis bandwidth and $g(u)$ is a bounded support function, first derivative of another bounded support function, $k(u)$, or kernel profile. It can be proven that, for a kernel with a convex and monotonically decreasing profile, the iterative procedure $x^{l+1} = m(x^l) + x^l$ converges.

All points $x_i, i = 1, \ldots, t_u$ belonging to a mode will converge to the same point, the mode center, or mean μ_u. Moreover, if we assume Gaussian modes, for each feature,

in this case the components of the colour vector $\boldsymbol{v}$, the p.d.f consists of a weight sum of the U modes modelled by a Gaussian distribution. A threshold test can simply be applied to the estimated p.d.f:

$$\sum_{u=1}^{U}\prod_{f=1}^{F}\omega_{(u,f)}\eta(x_f,\mu_{(u,f)},\sigma^2_{(u,f)}) < T_{MS}$$

Note that we are assuming that the features $f, f = 1,\ldots,F$ are independent. The weights $\omega_{(u,f)}$ also need to be estimated, and are generally defined by heuristics. If the probability estimated for a given pixel value $\boldsymbol{v}$ is smaller than the threshold T, the pixel is classified as foreground. The method and optimizations proposed in [11] were implemented and tested. The mean shift algorithm was tested for all sequences with the parameters: $N = 50$, $h = 3$ and $T_{MS} = 10^{-20}$.

6) Comparison – A comparative study of background modelling techniques was previously presented in [10], however this study consists of a theoretical comparison of several algorithms and no qualitative tests are presented. In order to get a better understanding of the algorithms, we tested them in several sequences. The results for the sequences SW, SH and OD are presented next. Please refer to section 3 for more details on the test sequences. All tests were executed using only colour vectors as features; the YUV colour space was used. The measure used to compare the segmentations and the ground-truth was the Perceptual Spatial Quality (PSpQ) measure proposed in [2]. All algorithms were implemented by the authors and the tests were performed in a Pentium 4 3.4GHz with 1GB of RAM.

Table 1 summarizes the results obtained for each algorithm in each sequence; for each algorithm-sequence combination the PSpQ measure and the frames per second (fps) are presented. No post-processing was employed on each algorithm's output segmentations. Figure 2 shows the evolution of PSpQ over time from the evaluated sequences.

		SW	SH	OD
RAvg	PSpQ	0.942	0.816	0.909
	fps	233	212	278
MoG	PSpQ	0.963	0.970	0.983
	fps	7.3	5.8	7.5
KDE	PSpQ	0.914	0.770	0.923
	fps	2.5	2.3	2.6
PF	PSpQ	0.823	0.723	0.800
	fps	4.1	2.9	3.9
MS	PSpQ	0.975	0.900	0.935
	fps	0.04	0.04	0.05

Table 1. Average PSpQ measure and frames per second over the evaluated frames of each sequence.

Results show that the mixture of Gaussians and mean shift algorithm perform consistently better than the others. However, the latter's processing time is extremely high, which invalidates its use for real-time applications. With these results in mind, we have opted to base our algorithm on the mixture of Gaussians approach to background modelling.

2.4. Summary

The algorithm can be seen as cascade of different techniques resulting in the refinement of the segmentation provided by the previous. First we use the statistical test (1) to determine the set $\mathcal{S}_a$ of pixels that are identified as being changed by any phenomenon other than noise - in our model a structural change or illumination variation. Then, on that set of pixels a simple collinearity test (2) is performed in order to assert that the modification was not due to some modification in illumination conditions, removing any pixel that are, resulting in a new set $\mathcal{S}_b$ of candidate pixels. Finally, the set is further refined eliminating any structural change that resulted from some repetitive dynamic behaviour of the background using the mixture of Gaussians modelling. The result is the final set of pixels $\mathcal{S}_c$ which are labelled as belonging to the foreground, i.e., any relevant moving object.

Although the different classification tests appear cascaded, some interaction happens between them. Namely, if the change results from camera noise and not from other factor, the model is updated with the current background value instead of the new frame value. This way we effectively reduce model error by only introducing modifications to the model when any other phenomenon than noise changes the pixel value. Also, background representation defined by the MoG model and described by (5) is used as the reference image $\mathcal{F}^r$.

Figure 3 summarizes the algorithm in pseudo-code, which until the end of the article will be called as cascaded mixture of Gaussians (CMoG).

3. Results and discussion

Several sequences were used to evaluate the proposed method performance. The first sequence, called shopping (SH) shows a view of a shopping corridor and is one of the test case scenarios made publicly available by the EC Funded CAVIAR project/IST 2001 37540 [1]. The scene consists of people walking, browsing the stores' displays or waiting for others. It has stable illumination conditions, except for a small portion in the right side of the field of view. However, hard shadows and reflections in the floor and in the display's glass are present. The second sequence,

[1] OneShopOneWait1cor sequence available at `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`

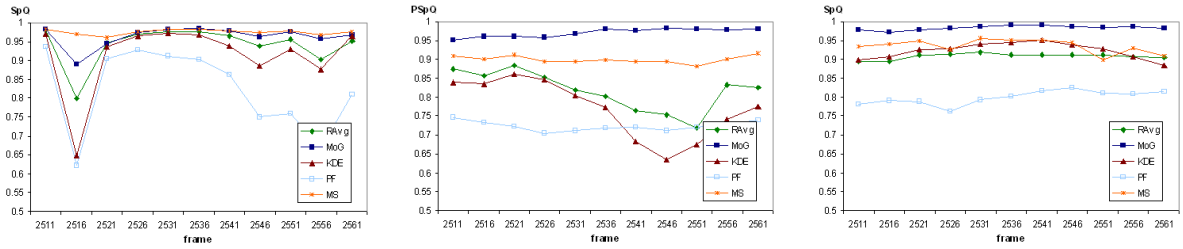

Figure 2. PSpQ for each tested frame in the sequence SW, SH and OD (from left to right).

```
Input: α, K, T_N, T_I and T_S
begin
  for t = 0, ...
    get current frame Fc
    foreach pixel in Fc
      // classification
      if (1) > T_N
        classify pixel as background
      else if (1) > 10^-10 T_N
        if (2) > T_I
          classify pixel as background
      else if (4) > T_S
          classify pixel as background
      else
          classify pixel as foreground
      // update background
      update ω_k, μ_k and σ_K in (4)
      update reference using (5)
    end foreach
  t = t + 1
  end for
end
```

Figure 3. Proposed algorithm.

labelled outdoor (OD) shows an outdoor scene with several people passing along the camera field of view and is available from the MPEG-7 test set (results are presented for stream A). The sequence has some noise and although the illumination conditions are fairly stable, the background presents significant vegetation swing. The speedway (SW) sequence was captured from a bridge over a speedway and is also available from the MPEG-7 test set (results are presented for stream 5). It shows different sorts of vehicles moving in both directions. Overall, it is the most stable stream regarding background changes but some relevant shadows are present. To compare the results, some frames of each sequence were manually segmented by visual inspection in order to obtain a ground-truth set. The following frames were considered: 350, 355,... and 400 in the SH sequence; 880, 885,... and 930 in the OD sequence; and 2510, 2515,... and 2560 in the SW sequence. Besides these three sequences, the test set also includes nine sequences first used in [8] and made available[2], namely: Meeting room with moving curtain (MR), Campus with wavering tree branches (CAM), Lobby in an office building with switching on/off lights (LB), Shopping center (SC), Hall of an airport (AP), Restaurant (BR), Subway station (SS), Water surface (WS), and Fountain (FT).

All experiments were performed using the YUV colour space. Moreover, to reduce the implementation complexity, it was considered a diagonal covariance matrix. The background model consisted of a mixture of 3 Gaussians, a learning rate α of 0.005 and a threshold T_S of 0.05. For the statistical test it was used a significance threshold T_N of 10^{-4}. Finally, for the collinearity test it was used a threshold T_I of 0.995. All thresholds were found through empirical testing to be fairly stable and can be used without modification for typical real-world scenes. The MoG algorithm was tested with the same parameters as in section 2.

Table 2 shows the average of two metrics from [2]: Perceptual Spatial Quality (PSpQ) and Relative Spatial Accuracy (RSpAcc). Results show that PSqQ present very similar results for both algorithms, since it tends to privilege larger segmentations. However, visual inspection of the segmentations show that the results can be very different as Figure 4 shows. This figure shows the segmentation results for frames 365 and 415 of the SH sequence, frame 600 of the OD sequence and frame 2550 of the SW sequence, from left to right. The top row shows the original frame, the second row shows the results from the MoG implementation of [7] and the third row shows the results obtained with CMoG. Taking this into consideration, RSpAcc was also used to compare both algorithms. In Figure 5 the evolution of the measure RSpAcc between the segmented frames and the "ground-truth" frames are presented.

The proposed method outperforms mixture of Gaussians modelling method in all test sequences. Even without post-

[2] http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

			SW	SH	OD	MR	CAM	LB	SC	AP	BR	SS	WS	FT
MoG method [7]	w/o pp	PSpQ	0.963	0.970	0.983	0.968	0.988	0.982	0.964	0.968	0.935	0.930	0.964	0.986
		RSpAcc	0.963	0.941	0.930	0.823	0.833	0.900	0.916	0.844	0.834	0.753	0.946	0.927
	w/ pp	PSpQ	0.938	0.953	0.978	0.961	0.982	0.970	0.918	0.946	0.889	0.838	0.956	0.983
		RSpAcc	0.975	0.947	0.970	0.911	0.940	0.961	0.950	0.905	0.860	0.789	0.973	0.973
CMoG method	w/o pp	PSpQ	0.941	0.971	0.984	0.966	0.988	0.981	0.951	0.972	0.922	0.943	0.962	0.985
		RSpAcc	0.978	0.971	0.976	0.967	0.898	0.956	0.968	0.953	0.921	0.852	0.972	0.980
	w/ pp	PSpQ	0.937	0.962	0.983	0.968	0.983	0.966	0.910	0.959	0.853	0.866	0.951	0.982
		RSpAcc	0.981	0.972	0.981	0.975	0.953	0.973	0.969	0.962	0.925	0.880	0.975	0.987

Table 2. Results over the selected sequences.

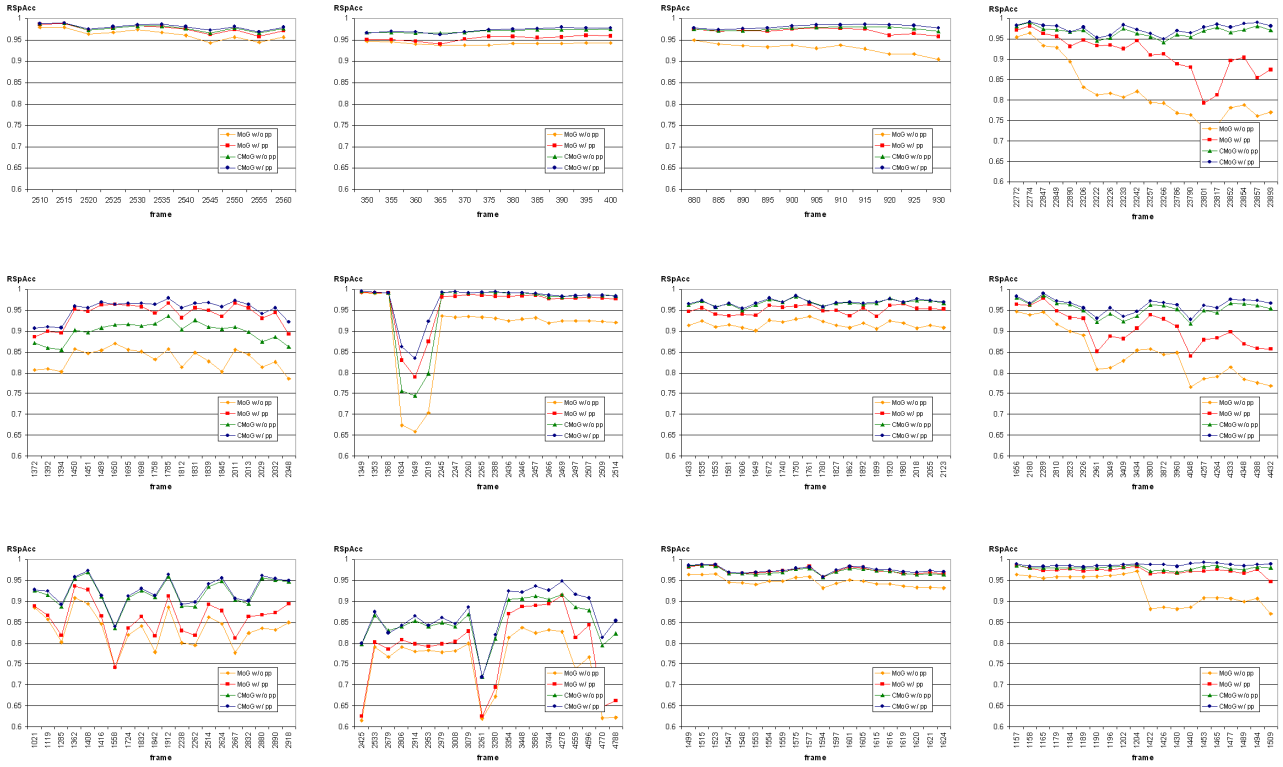

Figure 5. Results obtained with the measure RSpAcc for each sequence. Both algorithms MoG and CMoG are being evaluated without and with post-processing. From left to right and from top to bottom, the graphs are from the sequences SW, SH, OD, MR, CAM, LB, SC, AP, BR, SS, WS and FT.

processing the CMoG method performs significantly better than the regular MoG with post-processing in many sequences and has similar performance in the other two sequences. In fact, for all sequences, CMoG's results with and without post-processing do not differ much. Note that for noisy and highly dynamic scenes, like OD, the regular method without post-processing has the worst results. Also note that for the LB and SS sequences a noticeable drop happens at some point in time. This is due to sudden changes of global illumination, that are not properly handled by pixel-wise estimation of the background, as is the case – higher level input would be needed in order to quickly adapt to the changes. Nevertheless, CMoG easily returns to the normal classification performance. Additionally, the proposed method is faster than the original MoG: for 176×144 sequences, MoG performs at 26fps, while CMoG performs at 30fps; for 352×288 sequences, MoG performs at 6fps, while CMoG performs at 8fps.

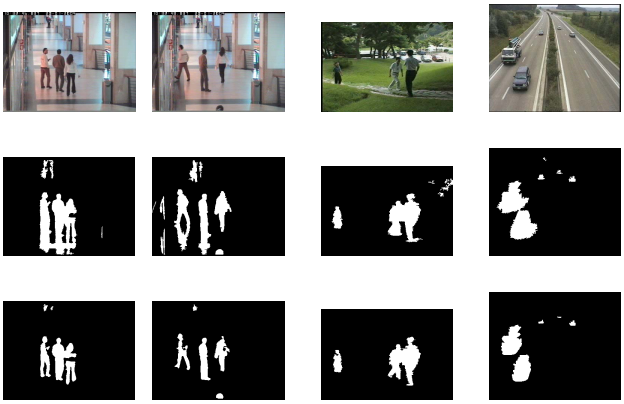

Figure 4. Segmentation results. The top row shows the original frame. The second row shows the results from the MoG implementation of [7]. The third row shows the results obtained with CMoG.

4. Conclusion

An efficient method of extracting moving objects (foreground) from a moderately dynamic background consists of modelling each pixel value evolution through time, by estimating the p.d.f. Several methods exist for this purpose, such as kernel density estimation, principal features statistical modelling, mean shift or mixture of Gaussians. An objective comparison was performed and results show that the latter proved to be the one that had the best performance/complexity relation. Additionally, we presented a method that performs a cascaded evaluation of typical dynamic elements that, although changing in time, we want to remove from the final foreground estimation. These elements include acquisition noise, illumination variation and repetitive structural changes or very slow in time. The proposed method performed consistently better than a regular mixture of Gaussians method. Even without post-processing the results show a similar or better performance than the regular method with post-processing. Moreover, typical illumination variation changes, like shadows, are successfully eliminated without the additional use of posterior complex shadow detection and suppression techniques.

Acknowledgments

This work has been partially supported by Fundação para a Ciência e a Tecnologia (Portuguese Science and Technology Foundation) and VISNET II, a Network of Excellence funded by the sixth Framework Programme of the European Commission.

References

[1] T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving video. *Signal Processing*, 31(2):165–180, March 1993.

[2] A. Cavallaro, E. D. Gelasca, and T. Ebrahimi. Objective evaluation of segmentation quality using spatio-temporal context. In *Proceedings of IEEE International Conference on Image Processing*, September 2002.

[3] A. Elgammal, D. Hardwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of European Conference on Computer Vision*, volume 2, pages 751–767, 2000.

[4] B. Han, D. Comaniciu, and L. Davis. Sequential kernel density approximation through mode propagation: Applications to background modeling. In *Proceedings of Asian Conference on Computer Vision*, 2004.

[5] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, August 2000.

[6] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proceedings of the European Conference on Computer Vision*.

[7] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):827–832, May 2005.

[8] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, November 2004.

[9] N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–1034 – I–1040, 2001.

[10] M. Piccardi. Background subtraction techniques: a review. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, October 2004.

[11] M. Piccardi and T. Jan. Mean-shift background image modelling. In *Proceedings of International Conference on Image Processing*, pages 3399–3402, 2004.

[12] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 19, pages 246–252, 1999.

[13] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proceedings of IEEE International Conference on Computer Vision*, volume 1, pages 255–261, 1999.

[14] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.