Background subtraction techniques: a review*

Massimo Piccardi
Computer Vision Group,
Faculty of Information Technology
University of Technology, Sydney (UTS), Australia
massimo@it.uts.edu.au

Abstract - *Background subtraction is a widely used approach for detecting moving objects from static cameras. Many different methods have been proposed over the recent years and both the novice and the expert can be confused about their benefits and limitations. In order to overcome this problem, this paper provides a review of the main methods and an original categorisation based on speed, memory requirements and accuracy. Such a review can effectively guide the designer to select the most suitable method for a given application in a principled way. Methods reviewed include parametric and non-parametric background density estimates and spatial correlation approaches.*

Keywords: background subtraction, moving object detection, parametric and non-parametric approaches, spatial correlation.

1 Introduction

Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called the "background image", or "background model". As a basic, the background image must be a representation of the scene with no moving objects and must be kept regularly updated so as to adapt to the varying luminance conditions and geometry settings. More complex models have extended the concept of "background subtraction" beyond its literal meaning.

Several methods for performing background subtraction have been proposed in the recent literature. All of these methods try to effectively estimate the background model from the temporal sequence of the frames. However, there is a wide variety of techniques and both the expert and the newcomer to this area can be confused about the benefits and limitations of each method. This paper provides a thorough review of the main methods (with inevitable exclusions due to space restrictions) and an original categorisation based on speed, memory requirements and accuracy.

The rest of the paper is organized as follows: Section 2 describes the main features of each method reviewed. Section 3 presents the comparison of speed, memory requirements and accuracy, in this order. Conclusive remarks are addressed at the end of this paper.

2 The reviewed approaches: from simple to complex

The approaches reviewed in this paper range from simple approaches, aiming to maximise speed and limiting the memory requirements, to more sophisticated approaches, aiming to achieve the highest possible accuracy under any possible circumstances. All approaches aim, however, at real-time performance, hence a lower bound on speed always exists. The methods reviewed in the following are:

- Running Gaussian average
- Temporal median filter
- Mixture of Gaussians
- Kernel density estimation (KDE)
- Sequential KD approximation
- Cooccurence of image variations
- Eigenbackgrounds

2.1 Running Gaussian average

Wren *et al.* in [1] have proposed to model the background independently at each *(i,j)* pixel location. The model is based on ideally fitting a Gaussian probability density function (pdf) on the last *n* pixel's values. In order to avoid fitting the pdf from scratch at each new frame time, *t*, a running (or on-line cumulative) average is computed instead as:

$$\mu_t = \alpha I_t + (1-\alpha)\mu_{t-1} \qquad (1),$$

where I_t is the pixel's current value and μ_t the previous average; α is an empirical weight often chosen as a trade-off between stability and quick update. Although not stated explicitly in [1], the other parameter of the Gaussian pdf, the standard deviation σ_t, can be computed similarly. In addition to speed, the advantage of the running average is given by the low memory requirement: for each pixel, this consists of the two parameters (μ_t , σ_t) instead of the buffer with the last *n* pixel values.

At each *t* frame time, the I_t pixel's value can then be classified as a foreground pixel if the inequality:

$$|I_t - \mu_t| > k\sigma_t \qquad (2)$$

holds; otherwise, I_t will be classified as background. The name *background subtraction* used to commonly indicate this set of techniques actually derives from (2).

Koller *et al.* in [2] remarked that the model in (1) is unduly updated also at the occurrence of such foreground values. For this reason, they propose to modify the model update as:

$$\mu_t = M\mu_{t-1} + (1-M)(\alpha I_t + (1-\alpha)\mu_{t-1}) \qquad (3),$$

where the binary value *M* is 1 in correspondence of a foreground value, and 0 otherwise. This approach is also known as *selective* background update.

As the model in [1] was proposed for intensity images, extensions can be made for multiple-component colour spaces such as (R,G,B), (Y,U,V), and others. Moreover, if real-time requirements constrain the computational load, the update rate of either μ or σ can be set to less than that of the sample (frame) rate. However, the lower the update rate of the background model, the less a system will be able to quickly respond to the actual background dynamic.

2.2 Temporal median filter

Various authors have argued that other forms of temporal average perform better than that shown in (1). Lo and Velastin in [3] proposed to use the median value of the last *n* frames as the background model. Cucchiara *et al.* in [4] argued that such a median value provides an adequate background model even if the *n* frames are sub-sampled with respect to the original frame rate by a factor of 10. In addition, [4] proposed to compute the median on a special set of values containing the last n_s sub-sampled frames and *w* times the last computed median value. This combination increases the stability of the background model.

The main disadvantage of a median-based approach is that its computation requires a buffer with the recent pixel values. Moreover, the median filter does not accommodate for a rigorous statistical description and does not provide a deviation measure for adapting the subtraction threshold.

2.3 Mixture of Gaussians

Over time, different background objects are likely to appear at a same *(i,j)* pixel location. When this is due to a permanent change in the scene's geometry, all the models reviewed so far will, more or less promptly, adapt so as to reflect the value of the current background object. However, sometimes the changes in the background object are not permanent and appear at a rate faster than that of the background update. A typical example is that of an outdoor scene with trees partially covering a building: a same (*i,j*) pixel location will show values from tree leaves, tree branches, and the building itself. Other examples can be easily drawn from snowing, raining, or watching sea waves from a beach. In these cases, a single-valued background is not an adequate model.

In [5], Stauffer and Grimson raised the case for a multi-valued background model able to cope with multiple background objects. Actually, the model proposed in [5] can be more properly defined an *image model* as it provides a description of both foreground and background values.

Stauffer and Grimson in [5] describe the probability of observing a certain pixel value, x, at time t by means of a mixture of Gaussians:

$$P(x_t) = \sum_{i=1}^{K} \omega_{i,t}\, \eta\left(x_t - \mu_{i,t}, \Sigma_{i,t}\right) \qquad (4),$$

with each of the K Gaussian distributions deemed to describe only one of the observable background or foreground objects. In practical cases, K is set to be between 3 and 5. Gaussians are multi-variate to describe red, green and blue values. If these values are assumed independent, the co-variance matrix, Σ_i, simplifies to diagonal. In addition, if the standard deviation for the three channels is assumed the same, it further reduces to a simpler $\sigma^2_i \boldsymbol{I}$.

For (4) to become a model of the background alone, a criterion is required to provide discrimination between the foreground and background distributions. In [5], it is given like this: first, all the distributions are ranked based on the ratio between their peak amplitude, ω_i, and standard deviation, σ_i. The assumption is that the higher and more compact the distribution, the more is likely to belong to the background. Then, the first B distributions in ranking order satisfying

$$\sum_{i=1}^{B} \omega_i > T \qquad (5),$$

with T an assigned threshold, are accepted as background.

At each t frame time, two problems must be simultaneously solved: a) assigning the new observed value, x_t, to the best matching distribution and b) estimating the updated model parameters. These concurrent problems can be solved by an expectation-maximisation (EM) algorithm working on the buffer of the last n frames. However, as this would prove extremely costly, the matching is approximated in these terms: amongst all distributions satisfying

$$\left(x_t - \mu_{i,t}\right)/\sigma_{i,t} > 2.5 \qquad (6),$$

the first in ranking order is accepted as a match for x_t. Furthermore, parameters $(\omega_{i,b}, \mu_{i,t}, \sigma_{i,t})$ are updated only for this matching distribution and by using simple on-line cumulative averages similar to that of (2). If no match is found, the last ranked distribution is replaced by a new one centered in x_t with low weight and high variance.

Amongst the many papers stemming from [5], that from Wayne Power and Schoonees is suggested to the reader as it elegantly describes the theoretical framework supporting the Stauffer-and-Grimson approach, while at the same time providing useful corrections [6].

2.4 Kernel Density Estimation

An approximation of the background pdf can be given by the histogram of the most recent values classified as background values. However, as the number of samples is necessarily limited, such an approximation suffers from significant drawbacks: the histogram, as a step function, might provide poor modeling of the true, unknown pdf, with the "tails" of the true pdf often missing. In order to address such issues, Elgammal *et al*. in [7] have proposed to model the background distribution by a non-parametric model based on Kernel Density Estimation (KDE) on the buffer of the last n background values. KDE guarantees a smoothed, continuous version of the histogram.

In [7], the background pdf is given as a sum of Gaussian kernels centered in the most recent n background values, x_i:

$$P(x_t) = \frac{1}{n} \sum_{i=1}^{n} \eta\left(x_t - x_i, \Sigma_t\right) \qquad (7).$$

Likewise (4), it seems to be dealing with a sum of Gaussians. However, differences are substantial: in (4), each Gaussian describes a main "mode" of the pdf and is updated over time; here, instead, each Gaussian describes just one sample data, with n in the order of 100, and Σ_t is the same for all kernels. If background values are not known, unclassified sample data can be used in their place; the initial inaccuracy will be recovered along model updates. Based on (7), classification of x_t as foreground can be straightforwardly stated if $P(x_t) < T$.

Model update is obtained by simply updating the buffer of the background values in fifo order by *selective* update (see Sect. 2.1): in this way, "pollution" of the model (7) by foreground values is prevented. However, complete model estimation also requires the estimation of Σ_t (which is assumed diagonal for simplicity). This is a key problem in KDE. In [7], the variance is estimated in the time domain by analysing the set of differences between two consecutive values.

The model proposed in [7] is actually more complex than what outlined so far. First, in order to address the issue of the time scale, two similar models are concurrently used, one for long-term and the other for short-term memory. Second, the long-term model is updated with a *blind* update mechanism so as to prevent undesired exclusion from the model of incorrectly classified background pixels. Furthermore, it addresses explicitly the problem of spatial correlation in the modeling of values from neighbouring pixel locations as described hereafter.

All the approaches at Sects. 2.1-2.3 model independently single pixel locations. However, it is intuitive that neighbouring locations will exhibit spatial correlation in the modeling and classification of values. To exploit this property, various morphological operations have been used for refining the binary map of the classified foreground pixels. In [7], instead, this same issue is addressed at the model level, by suggesting to evaluate $P(x_t)$ also in the models from neighbouring pixels and use the maximum value found in the comparison against T.

2.5 Sequential Kernel Density approximation

Mean-shift vector techniques have recently been employed for various pattern recognition problems such as image segmentation and tracking [8, 9]. The mean-shift vector is an effective gradient-ascent technique able to detect the main modes of the true pdf directly from the sample data with a minimum set of assumptions (unlike the Stauffer-and-Grimson approach, the number of modes is unrestricted). However, it has a very high computational cost since it is an iterative technique and it requires a study of convergence over the whole data space. As such, it is not immediately applicable to modeling background pdfs at the pixel level.

There have been recent approaches trying to solve this problem. In [10], Piccardi and Jan propose some computational optimisations promising to mitigate the computational drawback. Moreover, in a recent paper from Han *et al*., the mean-shift vector is used only for an off-line model initialisation [11]. In this step, the initial set of Gaussian modes of the background pdf is detected from an initial sample set. The real-time model update is instead provided by simple heuristics coping with mode adaptation, creation, and merging. In their paper, Han *et al*. compared the pdf obtained with their method against that of a KDE approach over a 500-frame test video, finding a low mean integrated squared error in the order of 10^{-4}; this justifies the name of *sequential Kernel Density approximation* (SKDA) that the authors gave to their method. Over the test video in [11], the number of modes showed to vary between 3 and 11, with an average of 8.

2.6 Cooccurrence of image variations

Seki *et al*. in [12] try to go beyond the idea of mere chronological averages by exploiting spatial cooccurrence of image variations. Their main statement is that neighbouring blocks of pixels belonging to the background should experience similar variations over time. Although this assumption proves true for blocks belonging to a same background object (such as an area with tree leaves), it will evidently not hold for blocks at the border of distinct background objects (this is likely the cause of several false detections shown in [12], Figs. 13-14, appearing at the borders of different background objects).

The method in [12] can be summarised as follows:

- instead of working at pixel resolution, it works on blocks of N x N pixels treated as an N^2-component vector. This trades off resolution with better speed and stability.

Learning phase:

- for each block, a certain number of time samples is acquired; the temporal average is first computed and the differences between the samples and the average are called the *image variations*;
- the N^2 x N^2 covariance matrix is computed with respect to the average and an eigenvector transformation is applied reducing the dimensions of the image variations from N^2 to K.

Classification phase for the current block:

- a neighbouring block, u, is considered, with its current input value; the corresponding current eigen image variation is computed, called z_u;
- the L-nearest neighbours to z_u in the eigenspace, $z_{(u,i)}$, are found and z_u expressed as their linear interpolation;
- the same interpolation coefficients are applied to the values of the current block, b, which have occurred at the same time of the $z_{(u,i)}$; this provides an estimate, z^*_b, for its current eigen image variation z_b;
- the rationale of the approach is that z_b and z^*_b should be close if b is a background block; to measure closeness, a cumulative probability over the 8-neighbouring blocks is used (the reader can refer to [11] for further details).

In [11], it is not specified whether the learning phase should be repeated over time to guarantee model update. As this model is based on variations, it is likely to show a natural robustness to limited changes in the overall illumination level. However, a certain update rate would be needed to cope with more extended illumination changes.

2.7 Eigenbackgrounds

The approach proposed by Oliver *et al*. in [13] is also based on an eigenvalue decomposition, but this time applied to the whole image instead of blocks. Such an extended spatial domain can extensively explore spatial correlation and avoid the tiling effect of block partitioning.

The method in [13] can be summarised as follows:

Learning phase:

- a samples of n images is acquired, each image with p pixels; the average image, μ_b, is then computed and all images mean-subtracted;
- the covariance matrix is computed and the best M eigenvectors stored in an eigenvector matrix, Φ_{Mb}, of size M x p.

Classification phase:

- Every time a new image, I, is available, it is projected onto the eigenspace as $I' = \Phi_{Mb} (I - \mu_b)$;
- I' is then back projected onto the image space as $I'' = \Phi^T_{Mb} I' + \mu_b$. Since the eigenspace is a good model for the static parts of the scene, but not for the small moving objects, I'' will not contain any such objects;
- Foreground points are eventually detected at locations where $| I - I'' | > T$.

The above procedure can be subject to variations improving its efficiency, but following a similar rationale. In [13], however, it is not explicitly specified what images should be part of the initial sample, and whether and how such a model should be updated over time.

3 Performance analysis

This section presents a comparative performance analysis based on speed, memory requirements and accuracy. Table 1 shows a synopsis of the results.

3.1 Speed

The fastest amongst the methods reviewed is certainly the Gaussian average, where, for each pixel, the classification is just a thresholded difference and the background model update adapts just one or two parameters. We define this time complexity as $O(1)$. The median filter has a similar classification cost, but model update can be approximated as linear in the number of samples, n_s (n_s is usually sub-sampled from the full sample set, n). The corresponding complexity can be stated as $O(n_s)$. The Mixture of Gaussians method has $O(m)$ complexity, with m the number of Gaussian distributions used, typically in the order of 3-5. For classifying a new pixel, the KDE model computes its value in the Gaussian kernels centered on the past n frames, thus raising $O(n)$ complexity, with n typically as high as 100. However, efficient implementation through the Fast Gauss transform can limit the actual execution time [14]. The model update has similar complexity, although it is likely to be performed at rates significantly lower than the frame rate. The SKDA method has $O(m + 1)$ complexity, where m is the number of modes of the approximated pdf. This value is not set a priori and depends on the actual data samples. However, in [11] this was shown to vary between 3 and 11 in a test video. The complexity for the cooccurrence-of-image-variations method can be estimated as $O(8 * (n + L^4 + L)/N^2)$, where n is accounted for searching the nearest neighbours amongst the n variations, L^4 is the estimated cost for computing the interpolation

coefficients, and L, that for applying them to the current block; amongst these, the dominant cost is either n or L^4. The N^2 denominator spreads the cost over the pixels in a block. However, the reader should be reminded that this method works at block instead of pixel resolution, and that the cost for updating the model has not been taken into account. Finally, the eigenbackground method has an estimated complexity per pixel of $O(M)$, where M is the number of the best eigenvectors. Here as well, possible costs associated with the model update have not ben considered.

Table 1. Background subtraction methods and performance analysis (refer to text for symbol explanation).

Method	**Spd**	**Mem**	**Acc**
Running Gaussian average [1,2]	*1*	*1*	*L/M*
Temporal median filter [3,4]	n_s	n_s	*L/M*
Mixture of Gaussians [5]	*m*	*m*	*H*
Kernel density estimation (KDA) [7]	*n*	*n*	*H*
Sequential KD approximation [11]	*m + 1*	*m*	*M/H*
Cooccurence of image variations [12]	$8n/N^2$	nK/N^2	*M*
Eigenbackgrounds [13]	*M*	*n*	*M*

3.2 Memory requirements

For some of the methods reviewed, the memory complexity per pixel is the same as the time complexity. Where this is intuitive, we will not enter into details. The memory complexity for the cooccurrence-of-image-variations can be estimated as $O(nK/N^2)$, where n is the number of variations in the training model and K their dimension. Again, the N^2 denominator spreads the cost over the pixels in a block. As the ratio K/N^2 is by definition largely less than 1, the estimated complexity turns out less than $O(n)$. At classification time, the eigenbackground method requires a memory complexity per pixel $O(M)$, with M the number of the best eigenvectors. However, at training time the method requires allocation of all the n training images, with an $O(n)$ complexity.

3.3 Accuracy

An extensive accuracy analysis is not possible in the scope of this paper, as it would require agreement on an experimental benchmark or a complex theoretical comparison. Here we limit the discussion to analyse the main model features and categorise each approach as providing limited, intermediate, or high (L, M, H) accuracy.

The methods with a background model based on a single scalar value can guarantee adaptation to slow illumination changes, but cannot cope with multi-valued background distributions. As such, they will be prone to errors whenever those situations arise. However, if such errors connect into small blobs, they can be removed from the classified image by an adequate size filter. Moreover, post-processing based on foreground object classification and tracking can recover errors performed at the background subtraction level.

For the approximation of a multimodal distribution, both parametric and non-parametric methods have been applied successfully. Consequently, both the Mixture of Gaussians and KDE approaches can model well the background pdf in general cases. In addition, in [7] the proposed KDE temporal model is complemented by a double time scale, spatial correlation and a combination of blind and selective update. These features are able to mitigate undesired artefacts such as ghosts and deadlocks.

Mean-shift methods can effectively model a multi-modal distribution without the need for assuming the number of modes a priori. However, their computational cost is very high. In the SKDA approach, they are used only in an initial stage. The model update is provided by heuristics for adapting, creating and merging the modes. [11] shows that SKDA proves a good approximation of a KDE model.

Examples of the accuracy achievable by the method based on the cooccurrence of image variations can be found in [12]. Differently from the other methods, this method works at block resolution in blocks of $N \times N$ pixels, thus limiting the

accuracy achievable at pixel level. In addition, blocks located at the border of different background objects might not exhibit significant cooccurrence thus risking to be misclassified.

We experimented the eigenbackground method with a training set with $n = 20$ recent images and $M = 3$ eigenbackgrounds. The quality of results was good but seemed to significantly depend on the images used for the training set. When the current image contained a moving object in the same position as in a training image, the projection in the eigenspace did not remove it completely. In [13], however, the authors report good results with lower computational load than a Mixture of Gaussians approach.

4 Conclusions

In this paper, we have presented a review of the most relevant background subtraction methods. This original review allows the readers to compare the methods' complexity in terms of speed, memory requirements and accuracy, and can effectively guide them to select the best method for a specific application in a principled way.

Amongst the methods reviewed, simple methods such as the running Gaussian average or the median filter offer acceptable accuracy while achieving a high frame rate and having limited memory requirements.

Methods such as Mixture of Gaussians and KDE prove very good model accuracy. KDE has a high memory requirement (in the order of a 100 frames) which might prevent easy implementation on low-memory devices. SKDA is an approximation of KDE which proves almost as accurate, but mitigates the memory requirement by an order of magnitude and has lower time complexity.

Methods such as the cooccurrence of image variations and the eigenbackgrounds explicitly address spatial correlation. They both offer good accuracy against reasonable time and memory complexity. However, practical implementation of the cooccurrence method imposes a trade off with resolution.

References

[1] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 19, no. 7, pp. 780–785, 1997.

[2] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell, "Towards Robust Automatic Traffic Scene Analysis in Real-time", Proc. ICPR'94, Nov. 1994, pp. 126-131.

[3] B.P.L. Lo and S.A. Velastin, "Automatic congestion detection system for underground platforms," Proc. of 2001 Int. Symp. on Intell. Multimedia, Video and Speech Processing, pp. 158-161, 2001.

[4] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 25, no. 10, pp. 1337–1442, 2003.

[5] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," Proc. IEEE CVPR 1999, pp. 246–252.

[6] P. Wayne Power and J. A. Schoonees, "Understanding background mixture models for foreground segmentation", in Proc. of Image and Vision Computing New Zealand 2002, Nov. 26-28, 2002, Auckland, New Zealand, pp. 267-271.

[7] A. Elgammal, D. Harwood, and L.S. Davis, "Non-parametric model for background subtraction," Proc. ECCV 2000, pp. 751-767.

[8] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 24, no. 5, pp. 603–619, May. 2002.

[9] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 25 no. 2 pp. 281–288, Feb. 2003.

[10] M. Piccardi, T. Jan, "Efficient mean-shift backgorund subtraction", to appear in Proc. of IEEE 2004 Int. Conf. on Image Processing, Singapore, 2004.

[11] B. Han, D. Comaniciu, and L.S. Davis, “Sequential kernel density approximation through mode propagation: applications to background modeling,” Proc. Asian Conf. on Computer Vision, 2004.

[12] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, “Background subraction based on cooccurrence of image variations,” Proc. CVPR 2003.

[13] N.M. Oliver, B. Rosario, and A.P. Pentland, “A Bayesian computer vision system for modeling human interactions,” *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 22, no. 8, pp. 831–843, 2000.

[14] A. Elgammal, R. Duraiswami, and L. S. Davis, “Efficient kernel density estimation using the fast Gauss transform with applications to color modeling and tracking,” *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 25, no. 11, pp. 1499–1504, 2003.