Improved Adaptive Gaussian Mixture Model for Background Subtraction

Zoran Zivkovic

Intelligent and Autonomous Systems Group
University of Amsterdam, The Netherlands
email: zivkovic@science.uva.nl

Abstract

Background subtraction is a common computer vision task. We analyze the usual pixel-level approach. We develop an efficient adaptive algorithm using Gaussian mixture probability density. Recursive equations are used to constantly update the parameters and but also to simultaneously select the appropriate number of components for each pixel.

1. Introduction

A static camera observing a scene is a common case of a surveillance system. Detecting intruding objects is an essential step in analyzing the scene. An usually applicable assumption is that the images of the scene without the intruding objects exhibit some regular behavior that can be well described by a statistical model. If we have a statistical model of the scene, an intruding object can be detected by spotting the parts of the image that don't fit the model. This process is usually known as "background subtraction".

A common bottom-up approach is applied and the scene model has a probability density function for each pixel separately. A pixel from a new image is considered to be a background pixel if its new value is well described by its density function. For a static scene the simplest model could be just an image of the scene without the intruding objects. Next step would be for example to estimate appropriate values for the variances of the pixel intensity levels from the image since the variances can vary from pixel to pixel. This single Gaussian model was used in [1]. However, pixel values often have complex distributions and more elaborate models are needed. Gaussian mixture model (GMM) was proposed for background subtraction in [2]. One of the most commonly used approaches for updating GMM is presented in [3] and further elaborated in [10]. These GMM-s use a fixed number of components. We present here an improved algorithm based od the recent results from [12]. Not only the parameters but also the number of components of the mixture is constantly adapted for each pixel. By choosing the number of components for each pixel in an on-line procedure, the algorithm can automatically fully adapt to the scene.

The paper is organized as follows. In next section we list some related work. In section 3 the GMM approach from [3] is reviewed. In sections 4 we present how the number of components can be selected on-line and to improve the algorithm. In section 5 we present some experiments.

2. Related work

The value of a pixel at time t in RGB or some other colorspace is denoted by $\vec{x}^{(t)}$. Pixel-based background subtraction involves decision if the pixel belongs to background (BG) or some foreground object (FG). Bayesian decision R is made by:

$$R = \frac{p(BG|\vec{x}^{(t)})}{p(FG|\vec{x}^{(t)})} = \frac{p(\vec{x}^{(t)}|BG)p(BG)}{p(\vec{x}^{(t)}|FG)p(FG)} \qquad (1)$$

The results from the background subtraction are usually propagated to some higher level modules, for example the detected objects are often tracked. While tracking an object we could obtain some knowledge about the appearance of the tracked object and this knowledge could be used to improve background subtraction. This is discussed for example in [7] and [8]. In a general case we don't know anything about the foreground objects that can be seen nor when and how often they will be present. Therefore we set $p(FG) = p(BG)$ and assume uniform distribution for the foreground object appearance $p(\vec{x}^{(t)}|FG) = c_{FG}$. We decide then that the pixel belongs to the background if:

$$p(\vec{x}^{(t)}|BG) > c_{thr}(= Rc_{FG}), \qquad (2)$$

where c_{thr} is a threshold value. We will refer to $p(\vec{x}|BG)$ as the background model. The background model is estimated from a training set denoted as $\mathcal{X}$. The estimated model is denoted by $\hat{p}(\vec{x}|\mathcal{X}, BG)$ and depends on the training set as denoted explicitly. We assume that the samples are independent and the main problem is how to efficiently estimate the

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

density function and to adapt it to possible changes. Kernel based density estimates were used in [4] and we present here an improvement of the GMM from [3]. There are models in the literature that consider the time aspect of an image sequence and the decision depends also on the previous pixel values from the sequence. For example in [5, 11] the pixel value distribution over time is modelled as an autoregressive process. In [6] Hidden Markov Models are used. However, these methods are usually much slower and adaptation to changes of the scene is difficult.

Another related subject is the shadow detection. The intruding object can cast shadows on the background. Usually, we are interested only in the object and the pixels corresponding to the shadow should be detected [9]. In this paper we analyze the only basic pixel-based background subtraction. For various applications some of the mentioned additional aspects and maybe some postprocessing steps might be important and could lead to improvements but this is out of the scope of this paper.

3. Gaussian mixture model

In practice, the illumination in the scene could change gradually (daytime or weather conditions in an outdoor scene) or suddenly (switching light in an indoor scene). A new object could be brought into the scene or a present object removed from it. In order to adapt to changes we can update the training set by adding new samples and discarding the old ones. We choose a reasonable time period T and at time t we have $\mathcal{X}_T = \{x^{(t)}, ..., x^{(t-T)}\}$. For each new sample we update the training data set $\mathcal{X}_T$ and reestimate $\hat{p}(\vec{x}|\mathcal{X}_T, BG)$. However, among the samples from the recent history there could be some values that belong to the foreground objects and we should denote this estimate as $p(\vec{x}^{(t)}|\mathcal{X}_T, BG+FG)$. We use GMM with M components:

$$\hat{p}(\vec{x}|\mathcal{X}_T, BG+FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \widehat{\vec{\mu}}_m, \widehat{\sigma}_m^2 I) \qquad (3)$$

where $\widehat{\vec{\mu}}_1, ..., \widehat{\vec{\mu}}_M$ are the estimates of the means and $\widehat{\sigma}_1, ..., \widehat{\sigma}_M$ are the estimates of the variances that describe the Gaussian components. The covariance matrices are assumed to be diagonal and the identity matrix I has proper dimensions. The mixing weights denoted by $\hat{\pi}_m$ are non-negative and add up to one. Given a new data sample $\vec{x}^{(t)}$ at time t the recursive update equations are [12]:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) \qquad (4)$$

$$\widehat{\vec{\mu}}_m \leftarrow \widehat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m \qquad (5)$$

$$\widehat{\sigma}_m^2 \leftarrow \widehat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T\vec{\delta}_m - \widehat{\sigma}_m^2), \qquad (6)$$

where $\vec{\delta}_m = \vec{x}^{(t)} - \widehat{\vec{\mu}}_m$. Instead of the time interval T that was mentioned above, here constant α describes an exponentially decaying envelope that is used to limit the influence of the old data. We keep the same notation having in mind that approximately $\alpha = 1/T$. For a new sample the ownership $o_m^{(t)}$ is set to 1 for the 'close' component with largest $\hat{\pi}_m$ and the others are set to zero. We define that a sample is 'close' to a component if the Mahalanobis distance from the component is for example less than three standard deviations. The squared distance from the m-th component is calculated as: $D_m^2(\vec{x}^{(t)}) = \vec{\delta}_m^T\vec{\delta}_m/\hat{\sigma}_m^2$. If there are no 'close' components a new component is generated with $\hat{\pi}_{M+1} = \alpha$, $\widehat{\vec{\mu}}_{M+1} = \vec{x}^{(t)}$ and $\hat{\sigma}_{M+1} = \sigma_0$ where σ_0 is some appropriate initial variance. If the maximum number of components is reached we discard the component with smallest $\hat{\pi}_m$.

The presented algorithm presents an on-line clustering algorithm. Usually, the intruding foreground objects will be represented by some additional clusters with small weights $\hat{\pi}_m$. Therefore, we can approximate the background model by the first B largest clusters:

$$p(\vec{x}|\mathcal{X}_T, BG) \sim \sum_{m=1}^{B} \hat{\pi}_m \mathcal{N}(\vec{x}; \widehat{\vec{\mu}}_m, \sigma_m^2 I) \qquad (7)$$

If the components are sorted to have descending weights $\hat{\pi}_m$ we have:

$$B = \arg\min_b \left(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f) \right) \qquad (8)$$

where c_f is a measure of the maximum portion of the data that can belong to foreground objects without influencing the background model. For example, if a new object comes into a scene and remains static for some time it will probably generate an additional stabile cluster. Since the old background is occluded the weight π_{B+1} of the new cluster will be constantly increasing. If the object remains static long enough, its weight becomes larger than c_f and it can be considered to be part of the background. If we look at (4) we can conclude that the object should be static for approximately $\log(1 - c_f)/\log(1 - \alpha)$ frames. For example for $c_f = 0.1$ and $\alpha = 0.001$ we get 105 frames.

4. Selecting the number of components

The weight π_m describes how much of the data belongs to the m-th component of the GMM. It can be regarded as the probability that a sample comes from the m-th component and in this way the π_m-s define an underlying multinomial distribution. Let us assume that we have t data samples and each of them belongs to one of the components of the GMM. Let us also assume that the number of samples that belong to the m-th component is $n_m = \sum_{i=1}^{t} o_m^{(i)}$ where $o_m^{(i)}$-s are defined in the previous section. The assumed multinomial distribution for n_m-s gives likelihood function

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

$\mathcal{L} = \prod_{m=1}^{M} \pi_m^{n_m}$. The mixing weights are constrained to sum up to one. We take this into account by introducing the Lagrange multiplier λ. The Maximum Likelihood (ML) estimate follows from: $\frac{\partial}{\partial \hat{\pi}_m} \left(\log \mathcal{L} + \lambda(\sum_{m=1}^{M} \hat{\pi}_m - 1) \right) = 0$. After getting rid of λ we get:

$$\hat{\pi}_m^{(t)} = \frac{n_m}{t} = \frac{1}{t} \sum_{i=1}^{t} o_m^{(i)}. \quad (9)$$

The estimate from t samples we denoted as $\hat{\pi}_m^{(t)}$ and it can be rewritten in recursive form as a function of the estimate $\hat{\pi}_m^{(t-1)}$ for $t-1$ samples and the ownership $o_m^{(t)}$ of the last sample:

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + 1/t(o_m^{(t)} - \hat{\pi}_m^{(t-1)}). \quad (10)$$

If we now fix the influence of the new samples by fixing $1/t$ to $\alpha = 1/T$ we get the update equation (4). This fixed influence of the new samples means that we rely more on the new samples and the contribution from the old samples is downweighted in an exponentially decaying manner as mentioned before.

Prior knowledge for multinomial distribution can be introduced by using its conjugate prior, the Dirichlet prior $\mathcal{P} = \prod_{m=1}^{M} \pi_m^{c_m}$. The coefficients c_m have a meaningful interpretation. For the multinomial distribution, the c_m presents the prior evidence (in the maximum a posteriori (MAP) sense) for the class m - the number of samples that belong to that class a priori. As in [12] we use negative coefficients $c_m = -c$. Negative prior evidence means that we will accept that the class m exists only if there is enough evidence from the data for the existence of this class. This type of prior is also related to Minimum Message Length criterion that is used for selecting proper models for given data [12]. The MAP solution that includes the mentioned prior follows from $\frac{\partial}{\partial \hat{\pi}_m} \left(\log \mathcal{L} + \log \mathcal{P} + \lambda(\sum_{m=1}^{M} \hat{\pi}_m - 1) \right) = 0$, where $\mathcal{P} = \sum_{m=1}^{M} \pi_m^{-c}$. We get:

$$\hat{\pi}_m^{(t)} = \frac{1}{K} (\sum_{i=1}^{t} o_m^{(i)} - c), \quad (11)$$

where $K = \sum_{m=1}^{M} (\sum_{i=1}^{t} o_m^{(t)} - c) = t - Mc$. We rewrite (11) as:

$$\hat{\pi}_m^{(t)} = \frac{\hat{\Pi}_m - c/t}{1 - Mc/t}, \quad (12)$$

where $\hat{\Pi}_m = \frac{1}{t} \sum_{i=1}^{t} o_m^{(t)}$ is the ML estimate from (9) and the bias from the prior is introduced through c/t. The bias decreases for larger data sets (larger t). However, if a small bias is acceptable we can keep it constant by fixing c/t to $c_T = c/T$ with some large T. This means that the bias will always be the same as if it would have been for a data set with T samples. It is easy to show that the recursive version of (11) with fixed $c/t = c_T$ is given by:

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + 1/t(\frac{o_m^{(t)}}{1 - Mc_T} - \hat{\pi}_m^{(t-1)}) - 1/t \frac{c_T}{1 - Mc_T}. \quad (13)$$

Since we expect usually only a few components M and c_T is small we assume $1 - Mc_T \approx 1$. As mentioned we set $1/t$ to α and get the final modified adaptive update equation

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) - \alpha c_T. \quad (14)$$

This equation is used instead of (4). After each update we need to normalize π_m-s so that they add up to one. We start with GMM with one component centered on the first sample and new components are added as mentioned in the previous section. The Dirichlet prior with negative weights will suppress the components that are not supported by the data and we discard the component m when its weight π_m becomes negative. This also ensures that the mixing weights stay non-negative. For a chosen $\alpha = 1/T$ we could require that at least $c = 0.01 * T$ samples support a component and we get $c_T = 0.01$.

Note that direct recursive version of (11) given by: $\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + (t - Mc)^{-1}(o_m^{(t)}(\vec{x}^{(t)}) - \hat{\pi}_m^{(t-1)})$ is not very useful. We could start with a larger value for t to avoid negative update for small t but then we cancel out the influence of the prior. This motivates the important choice we made to fix the influence of the prior.

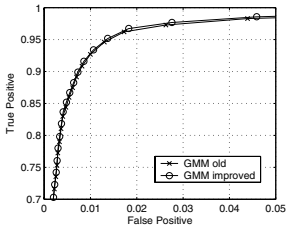

Figure 1. ROC curve for the laboratory sequence

5. Experiments

To analyze the performance of the algorithm we used three dynamic scenes. The sequences were manually segmented to generate the ground truth. We compare the improved algorithm with the original algorithm [3] with fixed number of components $M = 4$. For both algorithms and for different threshold values (c_{thr} from (2)), we measured the

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

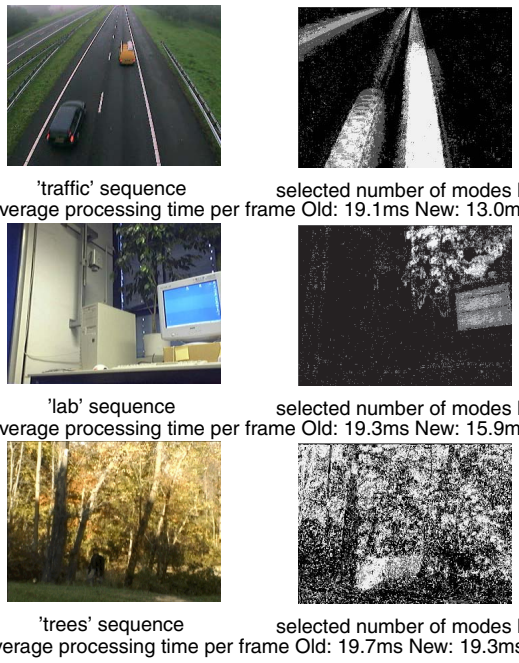

Figure 2. Full adaptation and processing times

true positives - percentage of the pixels that belong to the intruding objects that are correctly assigned to the foreground and the false positives - percentage of the background pixels that are incorrectly classified as the foreground. In figure 1 we present the receiver operating characteristic (ROC) curve for the 'lab' sequence. We observe slight improvement in segmentation results. The same can be noticed for the other two sequences (ROC curves not presented here). Big improvement can be observed in reduced processing time, figure 2. The reported processing time is for 320×240 images and measured on a 2GHz PC. In figure 2 we also illustrate how the new algorithm adapts to the scenes. The gray values in the images on the right side indicate the number of components per pixel. Black stands for one Gaussian per pixel and a pixel is white if maximum of 4 components is used. For example, sequence 'lab' has a monitor with rolling interference bars in the scene. The plant from the scene was swaying because of the wind. We see that the dynamic areas are modelled using more components. Consequently, the processing time also depends on the complexity of the scene. For the highly dynamic 'tree' sequence [4] the processing time is almost the same as for the original algorithm [3]. Intruding objects introduce generation of new components that are removed after some time (see 'traffic' sequence). This also influences the processing speed.

6. Conclusions

We presented an improved GMM background subtraction scheme. The new algorithm can automatically select the needed number of components per pixel and in this way fully adapt to the observed scene. The processing time is reduced but also the segmentation is slightly improved.

Acknowledgments

The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

References

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

[2] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," *In Proceedings Thirteenth Conf. on Uncertainty in Artificial Intelligence*, 1997.

[3] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *In Proceedings CVPR*, pp. 246–252, 1999.

[4] A. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric background model for background subtraction," *In Proceedings 6th ECCV*, 2000.

[5] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," *In Proceedings of ICCV*, 1999.

[6] J. Kato, S. Joga, J. Rittscher, and A. Blake, "An HMM-based segmentation method for traffic monitoring movies," *IEEE Trans. on PAMI*, vol. 24, no. 9, pp. 1291–1296, 2002.

[7] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models," *In Proceedings of ECCV*, 2002.

[8] P. J. Withagen, K. Schutte, and F. Groen, "Likelihood-based object tracking using color histograms and EM," *In Proceedings ICIP, USA*, pp. 589–592, 2002.

[9] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Formulation, algorithms and evaluation," *IEEE Trans. on PAMI*. vol. 25, no. 7, pp. 918–924, 2003.

[10] E.Hayman and J. Eklundh, "Statistical Background Subtraction for a Mobile Observer", *In Proceedings ICCV*, 2003.

[11] A.Monnet, A.Mittal, N.Paragios and V. Ramesh, "Background Modeling and Subtraction of Dynamic Scenes", *In Proceedings ICCV'03*, pp.1305–1312, 2003.

[12] Z.Zivkovic and F.van der Heijden, "Recursive Unsupervised Learning of Finite Mixture Models", *IEEE Trans. on PAMI, vol.26., no.5, 2004*.