TIME DEPENDENT ON-LINE BOOSTING FOR ROBUST BACKGROUND MODELING

Helmut Grabner, Christian Leistner and Horst Bischof
Institute for Computer Graphics and Vision, Graz University of Technology, Austria

Keywords: On-line learning, boosting, background modeling.

Abstract: In modern video surveillance systems change and outlier detection is of highest interest. Most of these systems are based on standard pixel-by-pixel background modeling approaches. In this paper, we propose a novel robust block-based background model that is suitable for outlier detection using an extension to on-line boosting for feature selection. In order to be robust our system incorporates several novelties for previous proposed on-line boosting algorithms and classifier-based background modeling systems. We introduce time-dependency and control for on-line boosting. Our system allows for automatically adjusting its temporal behavior to the underlying scene by using a control system which regulates the model parameters. The benefits of our approach are illustrated on several experiments on challenging standard datasets.

1 INTRODUCTION

For most video surveillance systems the detection of moving or intruding objects is of crucial importance. As they are easy to implement and fast to compute, frequently simple background subtraction methods are applied where objects are detected by blobs of pixels which do not correspond to the background model.

In its simplest form the background model (BGM) is solely one image, called the background image. Having this image, pixels are marked as foreground if they do not fit to it, i.e., more than a certain threshold above or below each pixel value. For realistic applications and in order to handle different environmental conditions (e.g., changing lighting conditions or foreground models moving to background and vice versa), more sophisticated, multimodal statistical models such as Mixture of Gaussians (GMM) (Stauffer and Grimson, 1999) or Eigenbackgrounds (Oliver et al., 2000) are often used. More efficient systems analyzing foreground models (Tian et al., 2005) have been proposed. In order to further improve the robustness some recent approaches exploit the spatial correlation between pixels arranged in blocks (e.g., (Russell and Gong, 2005)). Similarly, by describing statistics within one block using features (Heikkilä and Pietikäinen, 2006), the entire block is decided to be either background or foreground, which significantly improves the detection reliability.

Furthermore, several adaptive methods for estimating a pixel-based BGM have been proposed which update the existing image with respect to the current input frame (e.g., (Lo and Velastin, 2001), approximate median filter (McFarlane and Schofield, 1995)). For block-based BGMs adaptiveness can be achieved by describing each block with an on-line classifier (Grabner et al., 2006). This, additionally, allows to adapt to continuous changes (e.g., illumination changes in outdoor scenes) while observing the scene.

Although being effective for describing highly dynamic scenes, yet, the main drawback (see also section 2) of this on-line learner-based BGM is that its update strategy is similar to a self-learning method since the model updates are directly based on its own classifier predictions and can therefore end up in a catastrophic state (i.e., the model drifts away). Due to this dependency on its own predictions, the model performs quite well for a relatively short period of time but finally tends to learn foreground objects very quickly without offering any control on its temporal behavior (see also Figure 1). Furthermore, the cumbersome update strategy is highly scene dependent and, therefore, has to be hand-tuned every now and then.

To sum up, previously proposed background mod-
Since we use on-line boosting as learning algorithm, it is more suitable for the tasks where the data generation process changes over time. Especially, for the last point the algorithm has to be adaptable to the data. For off-line learning all labeled training samples have to be given in advance. Contrary, on-line algorithms can be hand-tuned but autonomously adapts to the underlying problem.

A weak classifier is a classifier which has to perform on a particular task with a certain tolerance/difficulty. This tolerance/difficulty may range from a few percentage points to a large margin. In general, boosting (see (Freund and Schapire, 1999)) extend the classifier-based background model by using an alternative version of the on-line learning algorithm which is controlled via temporal aspects and yield stable results over a long period of time. Furthermore, our proposed algorithm is resistant to outliers. Furthermore, our proposed algorithm is resistant to outliers. Choosing an alternative version of the on-line learning algorithm which is controlled via temporal aspects and yield stable results over a long period of time.

Section 2 extends the on-line boosting approach to temporal aspects while in section 3 we illustrate how these classifiers can be used for modeling background. Section 4 we (a) show illustrative examples and (b) evaluate the background model on public available datasets.

Three main ideas for further research.

1. A weak learning algorithm into a strong one. Therefore, given learning algorithm. In fact, boosting converts a machine learning for improving the accuracy of any weak classifier and for selecting one of those.
Concerning robust adaptiveness, this approach has several limitations:

First, the sample weights contribute forever to the entire model statistics and are only unlearned (i.e., getting less important) by updating the system with new ones. Hence, there is yet no control how fast information “fades” away and new one is gained.

Second, samples with high error get a much higher weight assigned than low-error samples. When sampling always from the same (static) distribution this is perfect, since boosting focuses on the difficult examples. Nevertheless, this makes the system extremely sensitive to label noise and, especially when dealing with an adaptive learning problem, this assumption makes no sense anymore.

2.2 Time Dependent On-line Boosting for Feature Selection

The first change considers the basic assumption in on-line boosting that the examples are all drawn from a fixed distribution. As we aspire fading memory we propose to use exponential forgetting of the examples over time. Therefore, we define the following update rule for the estimation of the value $\hat{w}_t$ using the previous value $\hat{w}_{t-1}$ and a new measurement $w$

$$\hat{w}_t = K \hat{w}_{t-1} + (1 - K) w$$

where $K = 2^{-\Delta t} \sqrt{0.5}$.\(^{(4)}\) gives the factor of how much previous information should be kept. In particular, it determines that half the information is kept in the time interval $\Delta t$. Once defined, this rule can now be easily used for all dynamic elements of the system. Especially, we incorporate this update rule for estimating the probability density functions (replacing the Kalman filtering like updates of mean and variance in Equation 3) of the weak classifier as well as the estimated errors, i.e., both $\lambda_{corr}$ and $\lambda_{wrong}$ in Equation 2.

The second change limits the effect of label noise. Each new incoming sample $\langle x_t, y_t \rangle$ for the first selector is initialized with the importance $\lambda = 1$. This means if it is well predictable its importance decreases by propagating through all the selectors otherwise increases. Assuming label noise a single noisy example can get very high importance and can change the entire model statistics rapidly. Therefore, we propose to keep the importance of the samples at the end of the ensemble constant. We first propagate the example through the set of selectors and obtain the value of $\lambda_n$ without doing an update. The actual update is done with the initial value set to $\lambda_1 = \lambda_n$ which clearly results in keeping it at one at the last selector. This simple modification ensures trusting the model more than the examples which means that the system, inherently, has some kind of outlier detection implemented. Please note, that this modification does not change the overall boosting process, it rather gives the example a prior importance.

Finally, we introduce a soft-selector, in order to limit the hard switching effect within the selectors. In contrast to taking the best weak classifier (i.e., that with the smallest error) for each selector it uses the information of all weak classifiers combined (which we have anyway). Although every arbitrary classifier fusion rule might be applied,
we chose to use the simple sum-rule which in practice yields good results (Kittler et al., 1998). As a result, all weak classifiers, the errors and therefore the importance weight $\lambda$ as well as the voting $\alpha_i$ are changing continuously.

3 BACKGROUND MODELING

First, we start with a review of the block-based classifier background model. Second, we show how to extend this approach by introducing a fixed and simple update strategy in combination with the extended time-dependent on-line boosting approach.

3.1 Classifier-based Background Model

In order to use classifiers for background modeling (Grabner et al., 2006), we partition the image into a grid of small, highly overlapping rectangular blocks (patches). For each of them a separate classifier is computed by combining simple image features (e.g., Haar features) which are selected using on-line boosting for feature selection. This is depicted in figure 2(a).

The main idea is to learn if the underlying image patch is predictable using the classifier. If so, this is considered as “allowed” background and otherwise it is considered as unknown (therefore foreground). Since boosting is used to train the classifier, which is a discriminative learning method, normally both positive and negative labeled samples are required in order to learn a decision boundary. Contrary, the task of background modeling is a one class classification problem, meaning only positive samples (the observed images) are given. Therefore, for each feature the negative distribution is estimated directly without learning (e.g., assuming each pixel is a random variable which is normally distributed or by using statistics of natural images (Huang and Mumford, 1999)). From this distribution simple learning rules are used to get a hypothesis for the weak classifier (for more details see (Grabner et al., 2006)).

The system has two phases: First, an initial learning stage where a separate classifier is built for all image patches assuming that all input images are positive examples (i.e., correspond to allowed background variations). Later on, in order to be adaptive to the scene, new input images are analyzed and the background model is updated according to a given, yet not totally traceable, policy. Then, it ends up with three different thresholds, which have to be hand-tuned as well as some higher update policy, for instance, that neighboring patches are inhibited to be updated for a certain time (yet another irreproducible variable) when the current patch is considered as foreground. Further on, due to its analogy to self-learning which relies on a direct feedback of its own predictions, the approach tends to drift and ends up in not predictable states when running for a long time (e.g., 24 hours a day, 7 days a week).

3.2 Robust Classifier-based Background Model

(a) classifier grid  
(b) self-learning  
(c) our approach: fixed-learning with adaptive parameters

Figure 2: (a) The background model is formed by a highly overlapping grid of classifiers. Updating the former approach (Grabner et al., 2006) uses self-learning (b). Our proposed method (c) does not directly take the classifier responses $t = C_{t-1}(x_t)$ into account and thus do not suffer from the drifting problem.

In order to get rid of the self-learning update strategy (Figure 2(b)), we propose fixed yet simple update strategy. Each classifier $C_i$ with the correspondent patch $x_i$, incorporates every new upcoming frame $t$ as a positive example $\langle x_i, t, +1 \rangle$.

For automatically updating the parameter $\Delta t$ we choose a simple dynamic control system, taking the following simple observation into account: The time constant should be large enough in order to model dynamic behavior but still as small as possible in order to be highly sensitive to small background changes. Please note, that of course one can also specify the time constant by hand in order to get a predefined background model for a specific application.
Figure 3: The first row depicts the test scene after $t = 100$ and the second row at $t = 251$, respectively. The second column shows the yielded confidences. As we assume the flickering monitor to be background, the confidences are quite high. In order to achieve such results our control system autonomously sets the $\Delta t$ in the flickering area higher than in the non-dynamic rest of the scene.

Changes. If observing a static scene then every movement or change should be considered as foreground. On the other hand, observing a dynamic behavior, e.g. leafs in the wind, this should be modeled, and therefore we have to increase the time constant up to a limit, where it is still possible to model these dynamic backgrounds. Furthermore, we assume that the time constant should move smoothly. For each individual on-line classifier having its own $\Delta t$ we use the following estimator

$$
\Delta t_t = K_i \Delta t_{t-1} + K_p \hat{\Delta} t,
$$

$$
\hat{\Delta} t = 1 - 0.5 \left(1 + \text{conf}(x)\right),
$$

where $K_i \in [0, 1]$ assumes the smoothness constraint and $K_p > 0$ is multiplied by the current estimate $\hat{\Delta} t$, which is considered to be proportional to the confidence of the patch $x$.

As soon as the confidence changes dramatically, i.e. a totally unknown intruder enters the scene, our control system tremendously increases $\Delta t$, which yields the implicit result that the harder the underlying scenario changes, the higher the controller sets $\Delta t$ and, thus, the longer it takes for a new object to "fade" into the background. Yet, smooth changes result in only small changes of $\Delta t$ which let the system adapt to small background changes, e.g. slightly changing lighting conditions.

This allows us to autonomously model different dynamic movements and periods for each classifier patch without drifting into unpredictable states since only $\Delta t$ and $\lambda_1$ as model parameters are changed but not the model itself (see Figure 2(c)).

The confidence of the classifier corresponds to the likelihood that the example correspond to the background. In fact, we robustly detect outliers and mark them as unknown foreground objects. Note, that all these would not be possible without the changes we proposed in the previous section.

4 EXPERIMENTS

In the following we demonstrate the benefits of our approach compared to existing methods. Therefore, we split the experiments into two main parts. First, we give a detailed evaluation on a sample sequence. Second, to show that we obtain state-of-the-art results the proposed method is applied on public available benchmark data sets.

For all of our experiments we use a classifier grid with a patch-size of $20 \times 20$ with an overlap of 75%. To compute the classifier we use only 15 selectors each using a set of 30 weak classifiers. Haar-like features are used because they can be evaluated very fast using the integral data structure. The thus obtained grid of detectors is evaluated and updated whenever a new frame arises. In order to set the time constant $\Delta t$ for each classifier we set $K_i = 0.95$ and $K_p = 10$.
Figure 4: Confidences (b) and estimated time-constants (c) for two different patches (a) over time. As can be seen, the blue patch covers a non-dynamic scene, reliably detects the intruder and immediately recovers its state. The second patch (red) has to handle dynamic flickering background. Hence, the time-constant also stabilizes around a higher value than for the blue patch. However, when the scene changed dramatically the time-constant raised very high making it very difficult for the intruded object to fade into the background.

Figure 5: Detection results of our method for the test sequences presented in (Toyama et al., 1999). The first column shows the initial frame of each sequence, second column the test frame and the third column the hand segmented ground truth. In the last three columns our results (the real valued confidence map, a binary segmentation achieved be zero thresholding as well as the time constants) are depicted. Note that the performance could be easily increased by analyzing the confidences more closely.

In addition, Figure 4 depicts some more detailed results of two specific patches. The color of the plots corresponds to the color of the rectangles. In particular, the red patch is located on the screen which is flicker during the scene. Therefore its time constant is automatically set to a higher value so that the classifying function produces results similar to the on-line learned classifier (high confidence). In the region where the screen is located, the time-constant is automatically set to a higher value. This allows the patches located around the screen to model the flickering while still being able to detect the intruding object.
In this paper we introduced controllable time dependency into on-line boosting. This was achieved by incorporating a simple automatic control system which allows for keeping the model statistics more robust. This results in both a higher noise invariance and the ability to adjust the model time-period. Together with a fixed yet simple update rule, more sophisticated methods like a mean shift-based clustering can be applied. Additionally, soft-switching selectors further allow us to smoothly adapt to new problems.

Moreover, the proposed classifier is able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene. This map provides the confidence map. This map provides the confidence that could be used to detect sudden changes in background maintenance. In (Toyama et al., 1999), a test set for evaluating background subtraction methods was presented. It consists of seven video sequences, each addressing different application fields. We compared using the test set and achieved the results shown in Fig. 5. In the following, we will concentrate on evaluating the improved system and acquired new one. Second, each new example is incorporated into the model with equal importance. Therefore, we get a controllable fading memory strategy in order to forget old information and acquire new one. The main advantage is that the classifier is able to handle the Light Switch problem since we are able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene.

4 CONCLUSIONS

In (Toyama et al., 1999) a test set for evaluating background subtraction was presented. It consists of seven video sequences, each addressing different application fields. We compared our results using the test set and achieved the results shown in Fig. 5. In the following, we will concentrate on evaluating the improved system and acquired new one. Second, each new example is incorporated into the model with equal importance. Therefore, we get a controllable fading memory strategy in order to forget old information and acquire new one. The main advantage is that the classifier is able to handle the Light Switch problem since we are able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene.

4.2 Wallflower Test Sequences

In (Toyama et al., 1999). Wallflower: Principles and practice of background subtraction (Stauffer and Grimson, 1999) we were able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene.

Moreover, the proposed classifier is able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene. This map provides the confidence that could be used to detect sudden changes in background maintenance. In (Toyama et al., 1999). Wallflower: Principles and practice of background subtraction (Stauffer and Grimson, 1999) we were able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene.

4.2 Wallflower Test Sequences

In (Toyama et al., 1999). Wallflower: Principles and practice of background subtraction (Stauffer and Grimson, 1999) we were able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene. This map provides the confidence that could be used to detect sudden changes in background maintenance. In (Toyama et al., 1999). Wallflower: Principles and practice of background subtraction (Stauffer and Grimson, 1999) we were able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene. This map provides the confidence that could be used to detect sudden changes in background maintenance. In (Toyama et al., 1999). Wallflower: Principles and practice of background subtraction (Stauffer and Grimson, 1999) we were able to describe it. Note, at the end of the sequence there is a correct breakdown by the confidence value, because a person enters the scene. This map provides the confidence that could be used to detect sudden changes in background maintenance.