Error Analysis of Background Adaption

Xiang Gao, T.E. Boult[*]
EECS Department Lehigh Univ.
[xig3 | tboult]@eecs.lehigh.edu

Frans Coetzee[†], Visvanathan Ramesh
Siemens Corporate Research Center
rameshv@scr.siemens.com

Abstract

Background modeling is a common component in video surveillance systems and is used to quickly identify regions of interest. To increase the robustness of background subtraction techniques, researchers have developed techniques to update the background model and also developed probabilistic/statistical approaches for thresholding the difference. This paper presents an error analysis of this type of background modeling and pixel labeling, providing both theoretical analysis and experimental validation. Evaluation is centered around the tradeoff of probability of false alarm and probability of miss detection, and this paper shows how to efficiently compute these probabilities from simpler values that are more easily measured. It includes an analysis for both static and dynamic background modeling. The paper also examines the assumptions of Gaussian and mixture of Gaussian models for a pixel.

Keywords: Surveillance, Background Modeling, Error Analysis, Markov Chain, Equilibrium, Mixture Gaussian, EM algorithm, ROC curve.

1 Introduction

Video surveillance is a well studied problem with both systems and new approaches still being developed, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. include background modeling, detection and foreground (a.k.a. target) versus background labeling. The numerous approaches just cited differ in the type of background model used, the procedure used to update the background, the detection process and the labeling/tracking process.

One can view most systems as having three separate phases of processing including **low level** (adaptive change detection, per pixel labeling, and background/threshold updating), **middle level** (connecting pixels via some forms of region growing or merging) and **high level** (more complex processes such as temporal-tracking or target recognition). Wallflower, [12], further divides its discussion of the low level processing into three levels: **pixel level** performs probabilistic predictions of the expected background; **region level** fills in homogeneous regions of foreground objects; **frame level** detects sudden, global changes in the image and may swap in better approximations of the background. While the region and frame levels are also very important, this paper concentrates on the pixel level. The approach does model the use of feedback from higher level processes but it does not model how the feedback is computed from or depends upon the low level results.

The primary goal of this paper is to show how to use statistic analysis to more formally set the parameters used in background maintenance methods. Existing work either openly admits to setting blending and thresholding parameters by hand, e.g. [11, 12], or more commonly, does not mention how they are set. In this paper, a formal approach is presented which defines parameters, for a given scenario, that will achieve a particular false alarm vs miss detection probability tradeoff. The secondary goal of the approach is to characterize the sensitivity and range of performance of background maintenance for a given scenario. The resulting parameters will, of course, still be dependent on the camera and scenario, but will be determined in a principled way from data samples.

In Section 2, this paper defines an input distribution model and background pixel models. A labeler model is also defined, which accounts for the feedback from higher level system processes. Section 3 derives the probability of false alarm and miss detection in terms of more directly computable terms. An Expectation-Maximization (EM) algorithm is used, in Section 4, to recover the model of the system input distribution. Section 5 introduces an equilibrium analysis that supports the efficient computation of the distribution of the background model. In Section 6, the analyses on real data verify the models and also show some of their limitations. Using these models and error analyses, any input distribution can be converted to ROC (Receiver Operation Characteristic) curves, which allow the end user to explicitly select the $p_{\mathrm{FA}}/p_{\mathrm{MD}}$ tradeoff (section 7). While ROC curves can be measured via large amounts of experimentation, model fitting and equilibrium analysis greatly reduce the amount of needed data collection and experimentation. In particular, in section 8, we show that much of the measured data is non-stationary, and that the data which is stationary is still not very Gaussian in nature. We then discuss how to generalize the approach to dynamic models, and the limitations of doing so.

[*] This work supported in part by Siemens Corporate Research center, and by ONR MURI contract N00014-95-1-0601.

[†] Dr. Coetzee is presently in NEC Research Institute, Princeton NJ. He can be reached by email at: coetzee@research.nj.nec.com

2 Distribution Model: previous work

In this paper, we focus our attention on the pixel level. It is the pixel level processing phase that makes the preliminary classifications of foreground (target) versus background and also handles adaptation to changing background. This phase does not include handling radical changes, e.g. turning lights off in a room, nor does it group pixels into regions. At this level, the pixel is considered independent of other pixels and for most of the paper we simply discuss a single pixel in image. We also assume that the model is appropriate, that if large lighting changes occurred some higher level processes would replace/modify our current background model.

An assumption that underlies in many background modeling approaches is that if each pixel resulted from a particular surface under particular lighting, a single Gaussian would be sufficient to model the pixel value while accounting for acquisition noise. Since different objects may project to the same image point (if scene points move) and lighting can change, some systems want to presume multiple models, e.g. a Mixture of Gaussians (MOG), per pixel. Existing systems usually set K, the number of Gaussians, in the range 2 to 5, [13, 14]. Furthermore, for computational reasons, the covariance matrix is assumed to be diagonal (i.e. uncorrelated). Obviously, for the special case $K = 1$, this gives us the traditional Gaussian model. We also note that a MOG can also handle, via approximation, the case when a single pixel's intensity uni-modal distribution is NOT well modeled by a single Gaussian.

Either way, to use a MOG model, we will also need to assume that each component satisfies a quasi-stationary criterion: the signal is flat fading, i.e. the change in pixel intensity value is slow compared to the update rate of our model. In particular we presume, for now, that the above equation for the background model will reach an approximate equilibrium before the input signal changes. For MOG models, we also presume the high-level labeling process will correctly tell us which part of the mixture to update. Let us briefly review previous work on background modeling and the standard approaches.

The P-finder system [1] uses a multi-class statistical model for the tracked objects, but the background model is a single Gaussian per pixel. A single Gaussian per pixel is easy to recover and this type of model is also used in many other systems. If the model is appropriate then thresholding based on the standard deviation is statistically well justified. Some simpler systems even ignore the formal modeling of standard deviation and simply track the mean or some other models of central tendency and use an ad-hoc thresholding process.

Other papers have stated that the use of a single "background" can limit robustness especially when viewing outdoor scenes with considerable clutter, e.g. [11, 13, 14], and these systems support multiple background models per pixel. One such model, used in [13, 14] is a **Mixture of Gaussians** (MOG). Given the input samples, a mixture of Gaussians is fit to it. The parametric form of the MOG distributions can then be used to classify pixels. In [11], a simpler form is used which tracks only the central values of the two primary distributions for a pixel, but the thresholding procedure is ad-hoc. These papers draw mostly on intuition and insight and do not present experiments to justify their multiple background model assumption.

There are two approaches for maintaining/updating the background model: multi-sample and per frame processing. A few approaches, e.g. [10, 13], gather many samples per pixel (i.e. many images) and then use the multiple samples to compute statistical models such as Gaussian, mixture of Gaussians or non-parametric model respectively. These methods require considerably more memory and processing and are more complex. They cannot be directly handled by the analysis presented herein.

For single "Gaussian" model, one only needs to compute the mean and variance. To allow the system to adapt to changing backgrounds, one needs to compute this over a window of time. While cheaper than other multi-sample techniques, true computation of a windowed (running) mean and standard deviation is still costly because it requires storage of K images (for an temporal window of size N). Since this approach is statistical in nature, if the input data matches the model assumption (i.e. Gaussian), setting the thresholds via the variance estimates is well understood. For these types of systems, the choice of N is the critical "blending" parameter. Larger N makes the system slow to change but better removes random fluctuations. Choosing N is directly amenable to the approach described herein, but to shorten the presentation it will not be pursued farther.

The per-frame processing approaches seek to compute an updated background model for each new frame. This approach is probably more common because it requires much less storage and much less computation. The basic idea is to update the background model via temporal blending. In this de-facto standard method for background maintenance, the model is maintained via:

$$B^{t+1} = (1 - \alpha)B^t + \alpha I^t \tag{1}$$

where B^t is the background pixel at time index t (it is a superscript not an exponent), I^t the new input pixel and α is the blending parameter that determines the speed of the forgetting old background information. Note that B^t is generally not the window mean but it is an estimate of the central tendency of the data over a window. Extending this blending approach to estimate the sample variance is more difficult and hence determining threshold(s) for detection is problematic. Selecting the global detection threshold G and the blending parameter α is the primary subject of this study.

In systems with multiple backgrounds, a separate (higher-level) process often determines which of the many background to update and then uses equation 1 to update only that model. The straight forward process we shall use in this analysis is to update the model which is closer to the input.

While this basic step is quite common in background subtraction, there are variations on the blending process, e.g.

systems that change α based on confidence of pixel being background (or Foreground) and variations that use all integer approximations to equation 1, e.g. [11]. If the camera system is not static, e.g. [7], the system will reinitialize its background model after each camera move and then begin blending. We note that the analysis herein, which presumes the system reaches a stable state, may be less applicable to such stop-and-stare systems.

2.1 Background Modeling Summary

For the sake of simplicity of discussion we presume a two background model[1] Let the primary background be $B_p^t(\phi)$, and the secondary background be $B_s^t(\phi)$. We assume the pixel intensity value is $I^t(\phi)$, where the pixel index is ϕ. For grayscale images $\phi = (u, v)$, for n-channel color it is $\phi = (u, v, c)$. Without loss of generality, we presume the input at the time $t-1$ was closest to the primary model $B_p^{t-1}(\phi)$.

Given these, we define

$$D_p^t(\phi) \;=\; I^t(\phi) - B_p^t(\phi) \quad (2)$$

$$D_s^t(\phi) \;=\; I^t(\phi) - B_s^t(\phi) \quad (3)$$

And define variable, q, as $q = \begin{cases} p & \text{if } |D_p^t(\phi)| \leq |D_s^t(\phi)| \\ s & \text{if } |D_p^t(\phi)| > |D_s^t(\phi)| \end{cases}$

and the negated form $\bar{q} = \begin{cases} p & \text{when } q = s \\ s & \text{when } q = p \end{cases}$

In some systems, the update depends on feedback from upper layers. In particular, we allow, for some process, to label the pixel ϕ as being in the target set T Target or in the Non-target set N. Then we can define the generalized update as:

$$B_q^{t+1}(\phi) = \begin{cases} [1-\alpha']B_q^t(\phi) + \alpha' I^t(\phi) & \phi \in T \\ [1-\alpha]B_q^t(\phi) + \alpha I^t(\phi) & \phi \in N \end{cases} \quad (4)$$

where α' may be (generally is) smaller than α. And the other term is updated as

$$B_{\bar{q}}^{t+1}(\phi) = B_{\bar{q}}^t(\phi) \quad (5)$$

Our error analysis will be dependent on the distributions of the states of the background models B_p and B_s. Note that for the special case of only one background model, if $\alpha' = \alpha$ and the distribution of I is $N(\mu, \sigma^2)$, with only a bit effort one can show that the distribution of B_p will be $N(\mu, \frac{\alpha}{2-\alpha}\sigma^2)$. In this case, we see that decreasing α produced a more narrow (peaked) distribution in B_p compared to the distribution of input I. Empirically, this general observation holds for MOG models as well although when the Gaussians overlap significantly the results are less intuitive.

2.2 Labeler Model

We note that most systems use considerable high level processing to clean up their initial foreground/background labeling. However all these systems still depend on a good initial labeling. The goal of this paper, is to study the error properties of this first stage labeling.

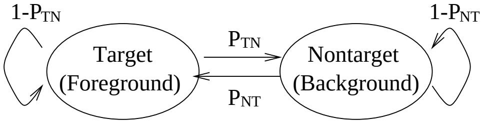


Figure 1. Two States Markov Model Labeler

In addition to updating our background model, we need to distinguish Non-target (Background) and Target (Foreground) pixels. We model the labeling of a pixel ϕ a Target pixel at time t if

$$T^t(\phi) = \begin{cases} q & D_q^t(\phi) > G^t(\phi) \\ 0 & otherwize \end{cases} \quad (6)$$

where $G^t(\phi)$ is the global threshold at time t. For now we will assume G is a constant. We also assume there is an ideal labeler $L^t(\phi)$, which provides the ground truth labels. We define a **false alarm** (FA) to be when $T^t(\phi) \neq 0 \& L^t(\phi) == 0$, and define a **miss detection** (MD) to be when $T^t(\phi) == 0 \& L^t(\phi) \neq 0$. Since we are considering only one pixel, in the remainder of the presentation we drop the explicit dependence on ϕ.

We note that the higher level processing can significantly impact the systems final FA/MD rates which can be quite different from the results presented here. Higher level processing generally decreases the FA while maintaining or slightly decreasing the MD rate. Still understanding the FA/MD rates here is critical to insure improving overall performance.

We also note that in systems with feedback, the update of the background model may depend on the output of the higher level labeling. Rather than define the details of the higher levels, which would make the behavior too particular to a given system, we introduce a high-level model for the labeler. We abstract away the details and use a simple Markov model of the higher level labeler's behavior. This model is sufficient to model the temporal (and non-independent) nature of "labeling errors". In most of the analysis we will further reduce the dependence on this high-level labeler model by assuming we "train" on target free data and then test using a mixture of targets and background.

Our model is a two-state "Markov Model Labeler", with two states shown in figure 1. This model has four transition probabilities that define its behavior. From this point of view, we consider it to be independent of the "data". For our analysis and simulation, we presume it has access to a data oracle (i.e. it knows the ground truth. We then define two parameters to summarize the labeler's work. They are P_{TN}, the probability of Labeler's output is non-target (background) given the ground truth is target, and P_{NT}, the probability of the Labeler's output is target given the ground truth is Non-target (background).

It is obvious that when both P_{TN} and P_{NT} are equal to 0, the "Markov Model Labeler" is an "Ideal labeler" which could exactly distinguish the pixel's class. If we set these two parameters to non-zero, then we can model errors in the

[1] Generalization of this two-mixture case to the general MOG case is straightforward but tediously complex to describe because the various mixtures must be reordered based on distance from the current value.

system's higher level classification. With error the transition probabilities of the finite state machine (FSM) will allow it to generate sequences of errors, although the sequence length is independent of actual error (such as a undetected lighting artifact).

To effectively use this model in the analysis, the transition probabilities can either be measured (by comparison with ground truth), and/or manually varied as a secondary parameter to see how the low level components are impacted by different higher level models.

3 Error Analysis

The error analysis can be divided into two parts. One is **Probability of False Alarm (p_{FA})**, the other is **Probability of Miss Detect (p_{MD})** . In the following analysis, we use the value at the current time t, B_p^t, B_s^t and I^t to calculate the state variables at the next time $t+1$, B_p^{t+1} and B_s^{t+1}. We let the L^t represent the output of the Labeler (not the oracle) at time t. By definition, B_p^t is dependent on I^{t-1}. Notice, however, that *the random variables B_p^t and B_s^t are independent of random variable I^t*. In other words, the pixels value we measure are not dependent on the current state of the system and the current state (as opposed to the next state) does not depend on the current measurement.

Suppose states variable $X^t = \left[B_p^t, B_s^t\right]^\top$ where the data set of B_p^t or B_s^t is $\{0, 1, \ldots, N-1\}$, and $\top$ is the transpose operator. Note are dealing with a discrete-time process $\{X^t : t \geq 0\}$ and a discrete state space, $\{0, 1, \ldots, J-1\}$.

Given the distribution of $p\{X_j^t\}$, we can get the p_{FA} and p_{MD}. To see this let $D_f^t(\phi) = \min\{|D_p^t(\phi)|, |D_s^t(\phi)|\}$ when $\min\{|D_p^t(\phi)|, |D_s^t(\phi)|\} > G$, or $D_f^t(\phi) = 0$. Then we have

$$\begin{aligned}
&p_{\text{MD}} \\
&= p\{L^t \in N | I^t \in T\} \qquad (7) \\
&= \sum_k \Big[\sum_j p\{L^t \in N | X_j^t, I^t = k, I^t \in T\} \\
&\qquad \cdot p\{X_j^t\}\Big] p\{I^t = k | I^t \in T\} \\
&= \sum_k p\{I^t = k | I^t \in T\} \cdot \Big\{ \sum_j p\{X_j^t\} \cdot \Big[\\
&\quad p\{L^t \in N | X_j^t, I^t = k, I^t \in T, D_f^t = 0\} \\
&\qquad \cdot p\{D_f^t = 0 | X_j^t, I^t = k, I^t \in T\} \\
&\quad + p\{L^t \in N | X_j^t, I^t = k, I^t \in T, D_f^t \neq 0\} \\
&\qquad \cdot p\{D_f^t \neq 0 | X_j^t, I^t = k, I^t \in T\}\Big]\Big\} \\
&= \sum_k \Big\{ \sum_j \Big[p\{D_f^t = 0 | X_j^t, I^t = k, I^t \in T\} \\
&\quad + P_{TN} \;\; p\{D_f^t \neq 0 | X_j^t, I^t = k, I^t \in T\}\Big] \\
&\qquad\qquad \cdot p\{X_j^t\}\Big\} p\{I^t = k | I^t \in T\}
\end{aligned}$$

Where we use the notation $I \in T$ to mean that the input pixel's value properly belongs to the target.

The sequence of deductions in equation 7 is just applications of *Bayes' Theorem* combined with the condition that when $D_f^t \neq 0$, the Labeler's output will always be B (background). Using a similar processes, we get

$$\begin{aligned}
&p_{\text{FA}} \\
&= p\{L^t \in T | I^t \in N\} \qquad (8) \\
&= \sum_j \Big[\sum_k p\{L^t \in T | X_j^t, I^t = k, I^t \in N\} \\
&\qquad \cdot p\{I^t = k | I^t \in N\}\Big] p\{X_j^t\} \\
&= \sum_k \Big\{ \sum_j \Big[P_{NT} p\{D_f^t \neq 0 | X_j^t, I^t = k, I^t \in N\}\Big] \\
&\qquad\qquad \cdot p\{X_j^t\}\Big\} p\{I^t = k | I^t \in N\}
\end{aligned}$$

To be able to use our error analysis, we need to estimate distribution of the background model $p\{X_j^t\}$ and input distributions $p\{I^t = k | I^t \in N\}$ and $p\{I^t = k | I^t \in T\}$. These values can be approximated from experimental measurements or, as we shall see in section 4, 5, the first of these can also be estimate via equilibrium analysis and the second and the third can be estimated via EM fitting.

4 EM algorithms

In this section, we discuss how to recover a model of the distribution of the input. It is obvious that the distribution that results from background maintenance is a function of input distribution.

We assume $p\{I^t = v\}$ is time-independent and use the normalized histogram of pixel intensity value as an approximation to the input distribution. We can then approximate that data with a mixture of K Gaussian models to estimate the parameters of input distribution. Like actual tracking systems that use MOG,[13, 14], we fit the MOG to the input using an Expectation-Maximization (EM) algorithm. Since EM fitting is a non-linear minimization process, for some of the input distributions the fitting may get stacked in a local minima.

In our analysis, we use the EM algorithm to fit for increasing values of K and stop when the new component has a "weight" less than 10^{-6}. When this occurs we leave the item blank in table 1, which generally show the mean absolute error between the normalized histogram and the MOG fitting.

Figure 2 shows some input histograms over time and in table 1, we list the fitting error of the output of EM algorithm [15] when we assume different component numbers in the mixture Gaussian distribution.

Figure 6, is the histogram of the dynamic models associated with the data shown in figure 2. The dynamic model is a histogram plots $B_q - I$. Table 2 shows the corresponding output of EM algorithm. From the table 1, 2 and figure 2, 6, one can draw some important conclusions:

- Even when there are no "targets", there is a much better fit (less than half as much residual error) with multiple Gaussian. For the MOG for MAE (A), the weight, mean and variance for the 1, 2, and 3 components are $0.9175 \cdot N(148.07, 10.3624), 0.0636 \cdot N(156.649, 9.83587)$ and

$0.01883 \cdot N(128.617, 954.443)$. Obviously the second and third components are being added in an attempt to handle the non-Gaussian nature of this data. Observations over a larger number of points show that adding second and third components for a single background generally add either a wide distribution centered at approximately the first model, or split a single peak into 2 adjacent peaks of similar magnitude.

- Fitting to the dynamic model, we could see that adding the second Gaussian component is still quite important. In table 2, significant decrease could be seen when we use large blending parameters, for small blending parameters, the gain is also larger than $\frac{1}{3}$. (This was non-intuitive and may be related to sampling artifacts).
- Generally, there is a significant improvement between the signal Gaussian and the two component mixture Gaussian model. And the result is acceptable for the general case. When we add the third and forth components into the two component Gaussian model, the gain is still noticeable but smaller. (As the number of terms grows so does the chance that EM algorithm is stuck in a local minimum point.)

5 Equilibrium Formula

Rather than running hundreds of experiments to develop the needed information we explore the use of equilibrium analysis. With this analysis, we simply start from a model of the input and use it to compute the distribution of the background model states (which depend on α, α' and the Markov labeler model parameters). To do this we assume the model for the labeler's behavior is will satisfy the Markov property. For technical reasons we must assume the labeler and background models are homogeneous discrete-time Markov chain that are aperiodic and irreducible. For a finite state space (e.g. integer values for X_j^t) this is satisfied and thus we know an equilibrium model, $p\{X_j^{\inf}\}$ always exists.

5.1 Computing $p\{X_j^{\inf}\}$

If we presume the input data is from a discrete stationary distribution, the distributions of these state variables can also be determined via equilibrium analysis, i.e. it is given by the distribution when $t \rightarrow \infty$. In order to get the equilibrium solution, we need to calculate the one step transition matrix, $[p\{X_j^{t+1} \mid X_i^t\}]$, of the state variables.

$$\begin{aligned} & p\{X_j^{t+1}|X_i^t\} \\ & \quad = \textstyle\sum_k \quad [p\{X_j^{t+1}|X_i^t, I^t = k, D_f^t = 0\} \\ & \qquad \cdot p\{D_f^t = 0|X_i^t, I^t = k\} \\ & \qquad + p\{X_j^{t+1}|X_i^t, I^t = k, D_f^t \neq 0\} \\ & \qquad \cdot p\{D_f^t \neq 0|X_i^t, I^t = k\}]p\{I^t = k\} \end{aligned} \tag{9}$$

Recalling equation 4, it is clear that for any given i and k $p\{X_j^{t+1}|X_i^t, I^t = k, D_f^t = 0\}$ is non-zero for for exactly 1 value of j.

The item $p\{X_j^{t+1}|X_i^t, I^t = k, D_f^t \neq 0\}$, is much more complicated because of the labeler's behavior.

$$\begin{aligned} & p\{X_j^{t+1}|X_i^t, I^t = k, D_f^t \neq 0\} \\ & \quad = \sum_{L^t} \{p\{X_j^{t+1}|L^t, X_i^t, I^t = k, D_f^t \neq 0\} \\ & \qquad p\{L^t|X_i^t, I^t = k, D_f^t \neq 0\}\} \end{aligned} \tag{10}$$

where L^t could be target (T) or Non-target (N).

Obviously, we can also compute

$$p\{I^t = k\} = \sum_{s \in (N \bigcup T)} p\{I^t = k|I^t \in s\}p\{I^t \in s\} \tag{11}$$

When we combine the above equations, we can finally estimate $p\{X^t\}$, the probability of background modeler states. The one step transition matrix depends on the inputs, the value of the blending parameters and the model of the labeler. We note that while this will save measurements, this process is not trivial. Computation of the equilibrium model currently takes a few hours. Fortunately, we can compute multiple models in parallel.

6 Input Histogram & MOG Fitting

We now look at model acquisition and verify how well the models fits the data. As we shall see, the results show that for these outdoor datasets with cameras in "automatic" mode, the use of a single Gaussian distribution is questionable. On pixels with no targets, a two term MOG model generally has half the modeling error of a Gaussian model.

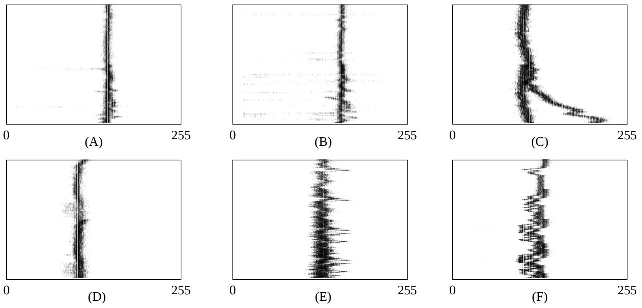


Figure 2. Histogram Intensity Over Time. Each row represents a histogram from 100 samples.

In figure 2, each picture shows the histogram of pixel intensity value of one pixel when the time is changing. Each row in a picture is one histogram, the darker pixels representing higher values. From the figure, we notice that the histogram is moving when the illumination is changing. For our input model, we accumulate these individual rows to provide a cumulative histogram for fitting.

In the figure 2 and 6, picture (A) is a pixel on the road where the target appear infrequently; picture (B) is also a pixel on the road, but the target appears more frequently; picture (C) is a pixel of a swaying leaf; picture (D) is a pixel on a waving short-grass area; picture (E) is a pixel on the wall of a building near the parking lots and picture (F) comes from shadow in the parking lots. Only figures (A) and (B)

MAE	Mixture Gaussian Model Errors ($\times 10^{-2}$)				
	One	Two	Three	Four	Five
(A)	5.65019	3.78010	3.32295	3.06526	
(B)	6.11431	2.59215	2.26253	2.26601	2.270
(C)	2.27294	0.76633	0.69196		
(D)	2.92134	2.59533			
(E)	0.78229	0.40811	0.40108		
(F)	1.11471	0.81691	0.78818		

Table 1. Fitting errors for EM Fitting of MOG to static model

contain any targets. From the table we see that in these six typical outdoor scenes, using more than two components in a mixture Gaussian distribution to calculate the background is not necessary, but using a single gaussian generally has a 15% to 200% larger error.

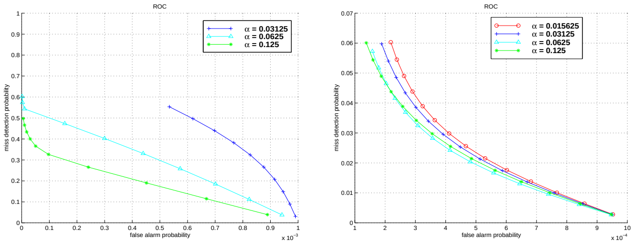
7 ROC curves

We have discussed the details of how to efficiently compute the p_{FA} and p_{MD}, let us look at how they could be used. We convert the p_{FA} and p_{MD} into ROC curves which can be used to set system parameters. To produce a ROC plot, all system parameters but one are fixed and a graph of p_{FA} vs p_{MD} is plotted as the parameter of interest is varied. One may combine multiple ROC plots for different values of some of the fixed parameters. In our case, the two parameters of most significant interests are the threshold G and the blending parameter α.

The user can then use these plots to set system parameters. Note that in all of the plots herein, the false alarm probabilities are plotted on a scale of $[0, 10^{-3}]$ because we are interested in very low rates of false alarms. As an alternative to human selection based on viewing graphs, we can automatically choose the parameters that meet some user specified optimization criterion, e.g. the minimum p_{MD} given that p_{FA} is no greater than 1 every minute. Implementing automated ROC analysis is trivial given the data needed for the plots.

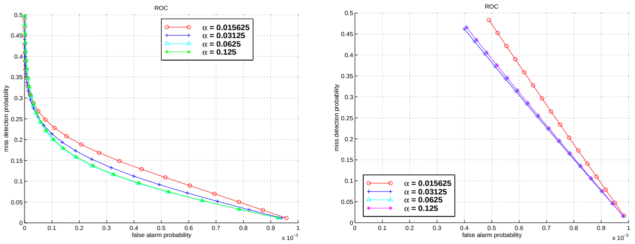
ROC curves/analysis have been used extensively for systems analysis and parameter setting. ROC analysis generally requires considerable experimentation and ground truth evaluation to the actual acquisition of the p_{FA} and p_{MD} data. The plots shown in this section usually require data from 100 “runs” for any set of input.

The proposed approach of developing models of the system’s behavior and deriving p_{FA} and p_{MD} in terms of simpler measurements, greatly simplifies their use for parameter setting for this low level vision task. By further splitting the modeling into background scenarios and target models, we further simplify the process. After annotation of a smaller number of sequences with targets, we can collect background models from more sequences where annotation is trivial because there are no targets. For the ROC analysis, we can analytically mix different “target” distributions and test them against different backgrounds. This also allows us to mix in targets that would be difficult to control in real experiments. For example, in figure 3 we have the similar background/target models. Both have a primary background described at $N(127.133, 5.60556)$. In the first case, we have 2 targets, one of which is broad and quite close to the background ($N(132.859, 98.256)$), with the second target distribution farther away ($N(72.0128, 159.729)$). In the second example, we simply relabel the $N(132.859, 98.256)$ as background. With training on many different inputs, we can have target models for pedestrians, cars, trucks and even targets which are trying to blend in (i.e. camouflaged targets).



(A) (B)

Figure 3. Linear plots of Static ROC curves with respect to different blending parameters. Left is 1 backround and 2 targets (one very close to background). Right is 2 background and 1 target. Note scales are $[0, 1]x[0, 10^{-3}]$ on the left and $[0, .07]x[0x10^{-4}]$ on the right. See text for discussion.



(A) (B)

Figure 4. Linear plots of Static ROC curves with respect to different blending parameters. (A) Static ROC for the pixel on the ground in a parking lot at CMU (figure 2(A)), (B) Static ROC for the pixel where leaf is swaying in the woods (figure 2(C)). Both plots have scales of $[0, .5]x[0, 10^{-3}]$.

When we change the plot method, figure 5, we could notice that even in the same curves, different blending parameters and different thresholds produce different effects in the different parts of the curves.

8 Dynamics vs statics

Up to this point, the error analysis has presumed that the input model and systems behavior are described by a set of stationary distributions. Of course we do expect lighting to change, but question how fast it changes and how much it will that impact the analysis. This section answers these questions and shows how to extend our analysis to handle the dynamic case.

If a distribution is stationary, then generating many sample histograms over varying sampling windows should pro-

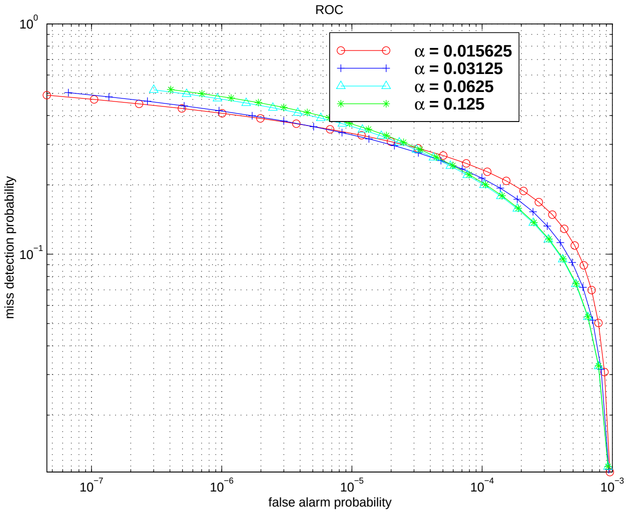


Figure 5. LogLog Plots of the ROC curves for the same data shown in the left of figure 4. The log plotting make it easier to see the separation of the curves at low p_{FA}.

duce, statistically, the same results. Simple observation of diagrams in section 6, for both indoor and outdoor scenes, shows that the dynamics is often significant. (Recall each row contains a histogram using 100 samples from the underlying distribution). Up to now, we have combined all that data and computed the expected distribution over longer periods, but that can significantly reduce system sensitivity.

We note that the definition/derivation of p_{FA} and p_{MD} did not depend on the distribution being stationary and hence the ROC analysis which is based on them, applies to the dynamic case as well. Of course, the distributions on which they are based are no longer stationary and hence both p_{FA} and p_{MD} are functions of time. Thus for a more meaningful analysis we would want to compute a statistics (mean/variance) on p_{FA} and mean p_{MD}. This generalization would allow us to combine measurements from different (but similar) pixels and scenarios, and collect data a over much wider range of conditions in a much shorter period of time.

The EM fitting to the data to recover MOG models still applies, as long as we make the temporal sampling fast enough so that each fitting is over an approximately stationary period. Note however, shorter sampling intervals decrease the accuracy of fitting.

The equilibrium analysis (EA), however, cannot be simply extended. The EA presumes that the background update equation has reached its stable state. If the background model is not stationary then it does not necessarily have an equilibrium. The EA is, however, simply a means of reducing our experimental/simulation effort and without it the analysis will be more expensive. We will have to gather data, either directly or via simulation, for various values of α.

Now recall that the use of temporal blending was not intended as an efficient way to approximate the mean, but rather as a technique to allow the system to track the background dynamics. Looking at the operation of the system, in fact, we are not as interested in the modeling of the background B_q as we are in modeling the statistics of $p\{B_q - I\}$. Thus we can ask the question: "For different values of α, how stationary is the distribution of $p\{B_q - I\}$?"

In figure 6 we show the histogram plots for $B_q - I$ for the same data show in figure 2(D). The plot shows α in the range $\frac{1}{8}$ to $\frac{1}{64}$. From this we can see that even in a case were there was significant scene dynamics, the temporal blending produces a distribution on $B_q - I$ is approximately stationary and hence we can proceed in the analysis using this dynamic model.

For this dynamic model we need to measure the input distributions, for each value of α (and α'). If this input and state distribution is stationary, then it again removes the need to compute p_{FA}, p_{MD} and the ROC curves over many samples. If α is very small, the distribution will not be as stationary as one might desire and then the collected model will be smeared. As before, such smeared model will be stationary but reduce sensitivity. While the distribution $p\{|B_q - I|\}$ requires considerably more experimentation to acquire, the computations for a few α values is inexpensive enough to be done simultaneously. Of course, we have still removed the need to measure the distributions for each value of T, but can now combine the values from different pixels so the number of experimental runs needed per scenario is still quite small.

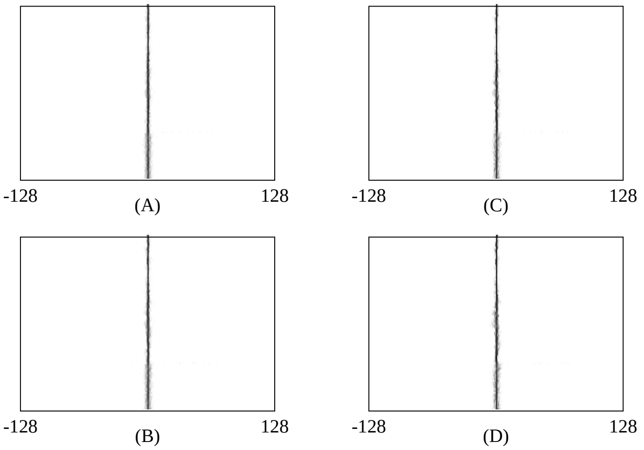
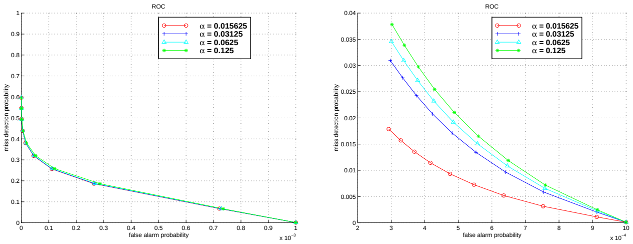


Figure 6. Dynamic model ($p\{B_q - I\}$). Each row is a histogram from 100 samples. (A) $\alpha = 0.125$, (B) $\alpha = 0.0625$, (C) $\alpha = 0.03125$, (D) $\alpha = 0.015625$

MAE	Mixture Gaussian Model Error ($\times 10^{-2}$)			
	One	Two	Three	Four
(A)	15.2057	0.54339		
(B)	15.5265	0.94054		
(C)	15.9132	9.84516	8.51713	
(D)	15.5553	10.1210	8.99075	

Table 2. Fitting errors for EM Fitting of MOG to the dynamic models shown in figure 7.

We also note that the derivations were in terms of B_q, which is the background model which is closer to I. In prac-



(A) (B)

Figure 7. Linear Plots of dynamic ROC curves with respect to different blending parameters using the same scenes shown in 3. Scale on left is $[0,1]x[0,10^{-3}]$, and on right is $[0,.04]x[2x10^{-4},10^{-3}]$

tice this is limiting in two ways. First, it means the update formula used to capture the model $p\{B_q - I\}$ will have to make the choice of which model is closer. This implies that some higher-level processes that produce the labels that are actually used in the update process will impact the measured distribution. It is important that the decision can be made independent of the global threshold G.

While a bit more expensive in terms of measurements the dynamic model allows better modeling of real scenes. It has the secondary advantage, in our experience, that the distribution $p\{B_q - I\}$ is much tighter, and hence provides for better sensitivity. While the distribution of $p\{B_q - I\}$ well centered, there is a temporal variation in its variance. With this observation it is not surprising that some systems use a threshold that is also dynamic. Given the dynamic model we can produce the ROC curves in figure 7 correspond to the static models in figure 3.

9 Conclusion

This paper has explored error analysis for background modeling and the primary change detection phase of video-based visual surveillance systems. The error analysis approach and basic equations can be adapted for analysis of a wide variety of self-adaptive change detection systems. The main contribution of the paper is showing how to reduce the number of experiments needed to generate ROC curves for this problem by their computation to combinations of simpler measurements and equilibrium analysis or input data model fitting. Using this process produces ROC curves which can be used to set system parameters.

As part of that experimental validation and EM model fitting to input data we also showed that, for the many different types of scenes considered, a single Gaussian per pixel was *NOT* a good model of the pixels value. The paper shows the problems with assuming, a static MOG model with white pixel "noise" and how a dynamic model can reduce these difficulties.

References

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

[2] B. Flinchbaugh and T. Olson, "Autonomous video surveillance," in *25th AIPR Workshop: Emerging Applications of Computer Vision*, May 1996. See also DARPA IUW May 1997.

[3] I. Haritaoglu, D. Harwood, and L. Davis, "w^4s: A real-time system for detecting and tracking people in 2.5d," in *Computer Vision—ECCV*, 1998.

[4] S. Intille, J. Davis, and A. Bobick, "Real-time closed-world tracking.," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 697–703, 1997.

[5] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[6] N. Friedman and S. Russell, "Image segmentation in video sequence: A probabilistic approach," in *Proc. of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 1997.

[7] A. Lipton, H. Fuijiyoshi, and R. Patil, "Moving target detection and classification from real-time video," in *Proc. of the IEEE Workshop on Applications of Computer Vision*, 1998.

[8] W. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 22–29, 1998.

[9] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background," in *FRAME-RATE Workshop*, IEEE, 1999. Eletronic (only) proceedings at www.eecs.lehigh.edu/FRAME.

[10] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *FRAME-RATE Workshop*, IEEE, 1999. Eletronic (only) proceedings at www.eecs.lehigh.edu/FRAME.

[11] T.E.Boult, R.Micheals, X.Gao, P.Lewis, C.Power, W.Yin, and A.Erkan, "Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets," in *Second IEEE International Workshop on Visual Surveillance*, pp. 48–55, IEEE, 1999.

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

[13] S. Rowe and A. Blake, "Statistical background modelling for tracking with a virtual camera," in *Proc. of British Machine Vision Conference*, 1995. Web version of a similar TR also availble.

[14] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246–252, IEEE, 1999.

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