12—1

Improved Adaptive Mixture Learning for Robust Video Background Modeling

Dar-Shyang Lee[1]
California Research Center
Ricoh Innovations, Inc.

Abstract

Gaussian mixtures are often used for data modeling in many real-time applications such as video background modeling and speaker direction tracking. The real-time and dynamic nature of these systems prevents the use of a batch EM algorithm. Currently, online learning of mixture models on dynamic data is achieved using an adaptive filter coupled with reassignment rules. However, convergence is very slow with a fixed learning rate typically employed in existing systems. In this report, we utilize an adaptive learning rate schedule to achieve fast convergence while maintaining adaptability of the model after convergence. Experimental results show a dramatic improvement in modeling accuracy using an adaptive learning schedule. Application of the proposed learning algorithm for video background modeling directly leads to improved approximation and robustness.

1 Introduction

Background subtraction is an early and essential processing component in many video analysis systems. Considering that point distribution observed at any pixel is often multimodal and time-varying in most real world applications, several researchers have proposed using Gaussian mixtures for pixel modeling [2],[3],[8]. The real-time and dynamic nature of the problem prevents the use of a batch EM [1] or an incremental EM [6] algorithm developed for mixture learning on stationary distributions that require storage of all observed data. These systems use an online learning algorithm that adapts to distribution changes via a recursive filter coupled with reassignment rules. However, the commonly used method of [8] converges very slowly. In this paper, we describe an improved method for online, adaptive mixture learning. We utilize an adaptive learning rate schedule to achieve fast convergence while maintaining model adaptability after convergence. Experimental results show a dramatic improvement in modeling accuracy using this technique and lead to a direct improvement in video background segmentation performance.

The rest of the paper is organized as follows. In Section 2, we first provide a brief description of existing methods and point out the fundamental difference in our approach. The proposed algorithm is then described in detail in Section 3. Experimental results on simulated data as well as real video data are reported in Section 4, followed by conclusions.

2 Related Works

Since the original proposal of using Gaussian mixtures for modeling pixel distributions in video [2], several researchers have followed the formulation of Stauffer and Grimson [8] for online learning of adaptive Gaussian mixtures. At each time step, parameters of the Gaussian that best matches the new observation $x(t)$ are updated using a recursive filter $\theta(t) = (1-\alpha)\cdot\theta(t-1) + \alpha\cdot\hat{\theta}(x;t)$ where α controls the temporal rate of adaptation. If x does not match the current model well, one of the Gaussian is reassigned to the new point. The weights for all Gaussians are also updated depending on whether it matches x and renormalized to 1. Through recursive filter learning and Gaussian reassignment, the system is able to model dynamic distributions.

The recursive low-pass filter learning is adequate for adapting to slow changes, but its convergence in the initial stage of parameter learning is unnecessarily slow. Consider the situation where a single Gaussian is assigned to learn a sequence of identical points. The low-pass algorithm initializes the mean to x and the variance to V_0. On subsequent iterations, since $x(t)-\mu(t) = 0$, $\sigma^2(t)$ would converge to 0 at a rate of $(1-\alpha)^t$. For a typical α value of 0.01, it takes approximately 460 iterations to reach 1% of V_0. It takes ten times as long for α =0.001. Therefore, the distribution must remain stationary for a long time to achieve adequate approximation. Although faster convergence can be achieved with a larger α , doing so would result in an unstable algorithm. With a low retention factor, the model will chase wildly after each new data point.

While recursive filter learning is necessary to track distribution changes, a more efficient strategy can be used to speed up convergence during initial parameter estimation. We point out that the learning rate for parameter estimation plays a different role from the retention factor controlling the adaptability of the model and, therefore, requires a different schedule that is adjusted through time. The former has the goal of achieving fast convergence on data distribution while the latter aims to adapt to slow migration in current data distribution and maintain model stability.

The same observation was made by other researchers [4], who proposed using expected sufficient statistics update equations at the initial learning stage to improve convergence, where the initial stage consists of the first L=1/α samples. After the initial stage, learning switches to a set of L-recent window update equations. By storing sufficient statistics of the first L samples in the early learning stage and applying the appropriate term weighting, convergence can be improved. However, this explicit division of learning stages can only be applied at initialization, when in fact subsequent Gaussian reassignment also suf-

[1] Address: 2882 Sand Hill Road, Suite 115, Menlo Park, CA94025, USA. E-mail: dsl@rii.ricoh.com

fers from slow convergence. In addition, we have found that the L-recent window update equations seem to be flawed and can lead to divergence.

The problem of selecting an appropriate learning rate schedule for online parameter estimation has been addressed in the literature. A detailed discussion on the necessary conditions for an effective learning schedule for stationary distribution can be found in [7]. It has been shown that with a $1/t$ schedule, the online EM algorithm can be considered a stochastic approximation to the batch EM algorithm and will converge to a local maximum of the likelihood function.

We propose a solution for fast online adaptive mixture learning by combining recursive filter learning with a $1/t$ learning rate schedule. To allow modeling of distribution changes, we use a modified $1/t$ schedule that converges to α instead of 0. In addition, since Gaussians are not updated with every x and may be reassigned, we apply a separate learning rate schedule to each Gaussian. Details of the proposed learning algorithm are described in the next section.

3 Algorithm Description

We developed a new algorithm by incorporating a modified adaptive schedule into the recursive filter learning. A counter is introduced for each Gaussian in the mixture to keep track of how many data points have contributed to the parameter estimation of that Gaussian. Each time the parameters are updated, a learning rate is calculated based on this counter, and the counter is incremented. When a Gaussian is reassigned, the counter is reset to zero. The learning rate is calculated such that it follows the basic $1/t$ schedule used in [7] when only few data points have been observed, and it approaches the same recursive filter learning in [8] after many points are seen. The counters effectively act as a separate clock for each Gaussian so that the appropriate learning rate can be used. At the cost of only one additional parameter per Gaussian, this modification dramatically improves the convergence while maintaining the same temporal adaptability after convergence.

For simplicity, we base our notation on a one-dimensional signal. Extension to a higher dimension is straight-forward. Let each pixel position in the video be represented by a K-Gaussian mixture

$$P(x) = \sum_{k=1}^{K} w_k \cdot g_k(x;\mu_k,\sigma_k) .$$

where g_k is the k-th normal distribution in the mixture. The Gaussians are initialized to 0 weights with a large variance V_0. Upon observing x at time $t+1$, the best matching Gaussian is determined by $\max\{p_k(x)\}$, where

$$p_k(x) = \hat{P}(G_k \mid x) = \frac{w_k \cdot g_k(x;\mu_k,\sigma_k)}{\sum_{j=1}^{K} w_j \cdot g_j(x;\mu_j,\sigma_j)} .$$

Since the normal distribution has infinite extent, $g_k(x)$ is usually rounded to 0 if x is more than three standard deviations away. For efficiency, a winner-take-all system sets the best matching $p_k(x)$ to 1, and all other p_j's to 0. The parameters of G_k are updated using the following

$$c_k = c_k + p_k(x)$$

$$\eta_k = p_k(x) \cdot (\frac{1-\alpha}{c_k} + \alpha)$$

$$\mu_k(t+1) = (1-\eta_k) \cdot \mu_k(t) + \eta_k \cdot x$$

$$\sigma_k^2(t+1) = (1-\eta_k) \cdot \sigma_k^2(t) + \eta_k \cdot (x-\mu_k(t))^2$$

If none of the Gaussian matches x, as is the case at initial learning, then one of the component G_k is selected and assigned to x by

$$\mu_k(t+1) = x \qquad \sigma_k(t+1) = V_0 \qquad w_k(t+1) = W_0 \qquad c_k = 1 .$$

The selection criterion can be based on $\min\{w_k\}$ or, as suggested in the context of background modeling [8], $\min\{w_k/\sigma_k\}$. With the exception of the reassigned Gaussian, the weights are updated using

$$w_k(t+1) = w_k(t) + \alpha \cdot (p_k(x) - w_k(t)) .$$

The weights are renormalized to sum to 1 after updates.

From the learning rate update equations above, it can be seen that in the initial learning stage of a Gaussian when only few samples have been observed, $\eta_k \approx 1/c_k$ and learning follows the basic $1/t$ schedule. Using the same example mentioned earlier where a constant x is observed, it takes only 55 iterations for σ to converge to 1% of V_0 when α=0.01, and 90 iterations when α=0.001, compared to 460 and 4600 iterations, respectively, with raw recursive learning.

As more data samples are included in its parameter estimation, η_k approaches α and behaves like the typical recursive learning. However, in contrast to the method of [4], there is no explicit separation of learning stages and the appropriate learning rate is applied at all stages, even through reassignment. Maintaining a separate counter for each Gaussian is not only necessary to account for reassignment, it also improves convergence and approximation of smaller data clusters.

We can qualitatively compare the proposed learning schedule to existing methods. The algorithm reported in [8] calculates the learning rate as $\eta_k = \alpha \cdot g_k(x;\mu_k,\sigma_k)$. Because $g_k(x)$ is usually very small, this makes convergence intolerably slow. This can be improved by simply using $\eta_k = \alpha$ without much side effect, as is done in [2][3]. This is the recursive filter learning with fixed learning rate we analyzed earlier, and serves as the basis for our comparison in the next section. In the method proposed in [4], $\eta_k = p_k(x)/c_k$ in the initial stage. After the first L samples, assuming L=1/α, the learning rule is equivalent to

$$\mu_k(t+1) = (1-\alpha) \cdot \mu_k(t) + \eta_k \cdot x$$

$$\eta_k = \alpha \cdot p_k(x) / w_k(t+1) .$$

However, since 1-α+η_k can exceed 1, the algorithm sometimes displays a diverging behavior.

4 Experimental Results

The proposed algorithm was tested on synthetic as well as real video data and showed remarkable improvement over existing algorithms. We used 3-Gaussian mixtures with α=0.001, a typical setting for systems reported in literature. In addition, V_0=1000 and W_0=α. We used $\min\{w_k\}$ as the selection criterion for Gaussian assignment. Input $\mathbf{x}$ is the pixel value in RGB or YUV space, ranges between 0 and 255 in each dimension. A diagonal covariance matrix was used.

We first tested the algorithms on several sets of synthetic data randomly generated from mixture distributions with known parameters. The results were very impressive. The proposed algorithm consistently out-performed the conventional fixed-rate recursive filter learning in both convergence speed and modeling accuracy through various parameter settings. Furthermore, since the convergence rate is almost independent of α, the performance of the proposed algorithm was almost unaffected by any reasonable choice of α. On the contrary, the conventional algorithm was very sensitive to the selection of α. Of course, the adaptability of both algorithms depends on α.

The proposed mixture learning algorithm was used in a system for video background segmentation [5] and led to significant improvement over the algorithm reported in [8]. Input video is obtained from a fixed camera positioned on a table in a meeting room. Figure 1 illustrates a very typical case of the difference between the algorithms. The value of the *red* color component of a pixel in the video is plotted over time with "+". The large cluster of constant values near the top of the graph, which is the color of the meeting room wall, is the background. Occasional deviations happen when a person moves in front of it. The shaded (green) region shows three standard deviations of the most dominant Gaussian in the mixture. The result from the fixed rate recursive filter learning is shown at the top, and the proposed algorithm shown at the bottom. It is apparent that the conventional algorithm converged very slowly on the wall color, even erroneously included some points from the person. As a result, all other points outside the shaded region are grouped in a second Gaussian (not shown). In contrast, the proposed algorithm quickly converged on the wall data. Almost all the wall data points are tightly enclosed in the shaded region. In addition, the other points are picked up by two other Gaussians: one tight cluster near the value of 70, and one with a wide spread covering between 100 and 200. Because the clusters are well approximated, the wall is easily detected as background.

The actual background segmentation results are shown in Figure 2. The person initially stood in front of the table (frame 0), moved about the room (frame 700), then grabbed the coffee mug and left the room near frame 1000 (not shown). Both algorithms were initialized using frame 0. Rows (a) and (b) show the background model and segmented foreground, respectively, obtained by the algorithm in [8]. It can be seen that remnants of the person from frame 0 was clearly visible at frame 700 and still vaguely visible even at frame 1500, leaving a ghost foreground region where the person stood initially. Similarly, the mug that was removed still left a ghost region 500 frames later. This is a direct consequence of the slow convergence on distribution changes based on fixed rate recursive filter learning. Results obtained from our algorithm, shown in rows (c) and (d), are dramatically improved. With fast adaptation and accurate modeling of data distribution, the system was able to quickly adapt to changes in the scene and detect the new background. In frame 700, only a very faint trace of the person remains in the background model, and the ghost region was largely eliminated. In frame 1500, the system quickly adjusted to the disappearance of the mug and once again avoided false foreground detection.

5 Conclusion

In this paper we proposed an algorithm that dramatically improves the convergence of online, adaptive mixture learning. This is achieved by incorporating a time-adaptive learning rate schedule developed for stationary distribution estimation into the recursive filter update equation for online mixture learning. Experimental results confirmed that this modification dramatically improves convergence and leads to better estimation. We showed that using the proposed algorithm in a video segmentation system based on the mixture model leads to a significantly better performance.

References

[1] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society B*, vol. 39, pp.1-38, 1977.

[2] Friedman, N. and S. Russell, "Image segmentation in video sequences: a probabilistic approach," in *Proc. of the 13th Conference on Uncertainty in Artificial Intelligence*, August, 1997.

[3] Harville, M., Gordon, G., and Woodfill, J. "Foreground segmentation using adaptive mixture models in color and depth," *ICCV Workshop on Detection and Recognition of Events in Video*, pp.3-11, 2001

[4] KaewTraKulPong, P. and Bowden, R. "An improved adaptive background mixture model for real-time tracking with shadow detection," *Proc. of 2nd European workshop on Advanced Video Based Surveillance Systems*, Sept. 2001.

[5] Lee, D-S, "A Bayesian framework for background segmentation based on adaptive Gaussian mixtures," to appear in Proc. *of 6th World MultiConference on Systemics, Cybernetics and Informatics*, Orlando, Florida, July 14-18, 2002.

[6] Neal, R. M. and Hinton, G. E. "A view of the EM algorithm that justifies incremental, sparse, and other variants", Learning in Graphical Models, 1998.

[7] Sato, M-A and Ishii, S. "Online EM algorithm for the normalized Gaussian network", *Neural Computation*, v.12, pp.407-432,1999.

[8] Stauffer, C. and Grimson, W.E.L., "Adaptive background mixture models for real-time tracking," *Proc. CVPR*, v.2, pp.246-252, June 1999.

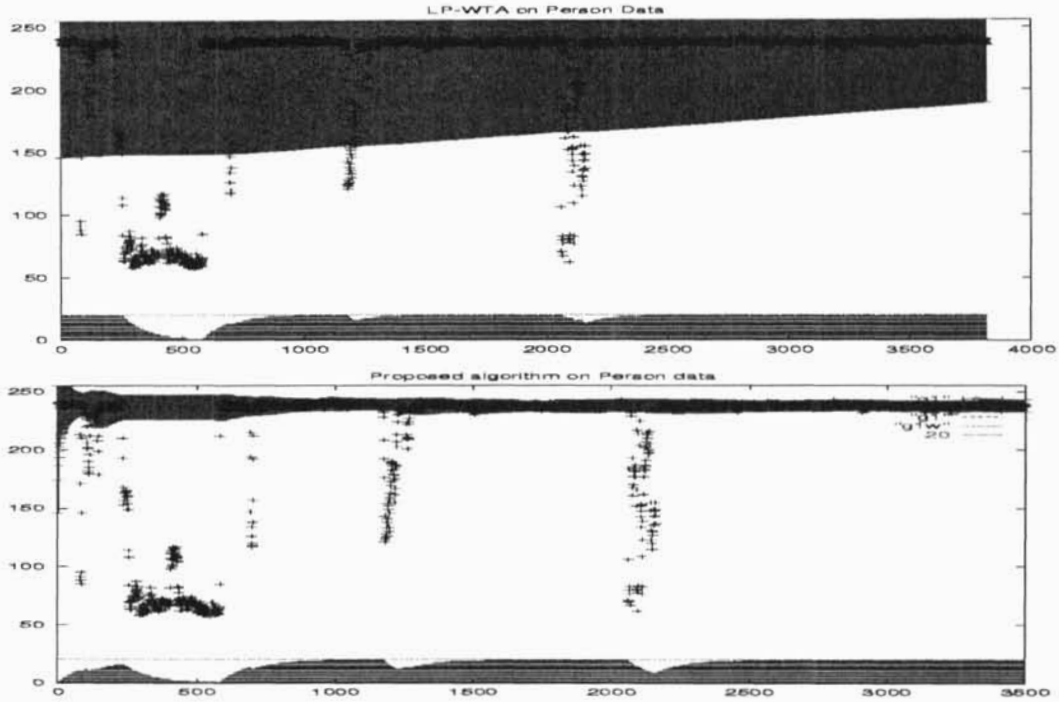

Figure 1 – A comparison of mixture learning algorithm in [8] (top) and the proposed algorithm (bottom). The value of the *red* color component (y-axis) of a pixel in video is plotted over time (x-axis) as "+". The shaded (green) region shows the estimated $(\mu \pm 3\sigma)$ for one of the three Gaussians in the mixture. The weight (magnified) of the Gaussian is shown at the bottom of each plot in pink.

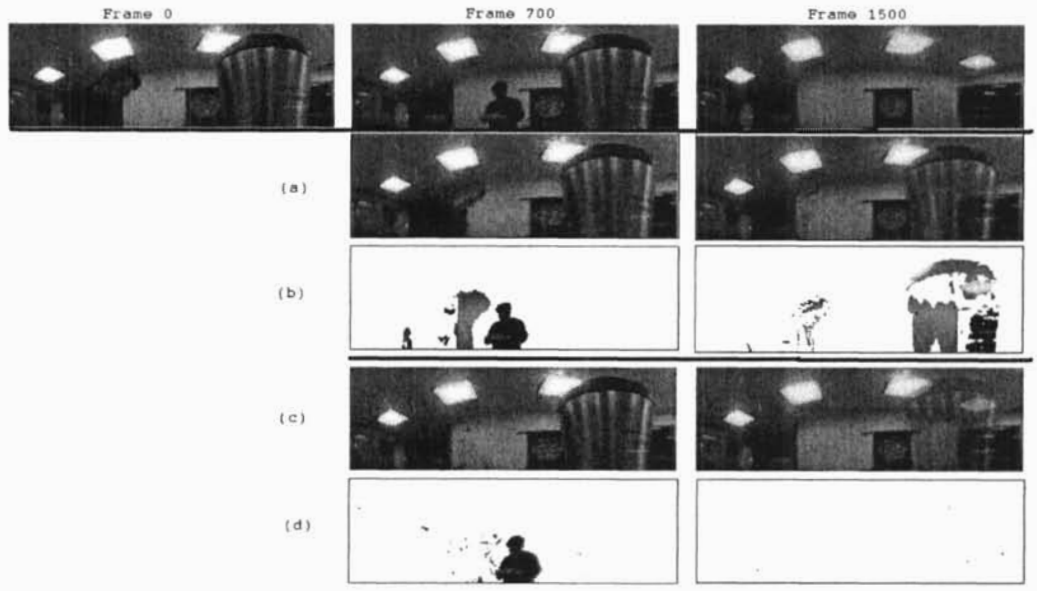

Figure 2 – A comparison of the proposed mixture learning algorithm used in video background segmentation. The top row shows frame 0, 700 and 1500 in the video. The person grabbed the coffee mug and left the room near frame 1000 which is not shown. Rows (a) and (b) show the background model and the segmented foreground region obtained by the typical fixed rate algorithm. A shadow of the person is clearly visible after 700 frames, causing a ghost foreground region. Rows (c) and (d) show the results obtained by the proposed learning algorithm. The model quickly adapted the disappearance of person and coffee mug and eliminated ghost foreground regions.