

BACTERIA-FILTERS: PERSISTENT PARTICLE FILTERS FOR BACKGROUND SUBTRACTION

Yair Movshovitz-Attias

Shmuel Peleg

School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

ABSTRACT

Moving objects are usually detected by measuring the appearance change from a background model. The background model should adapt to slow changes such as illumination, but detect faster changes caused by moving objects. Particle filters do an excellent task in modeling non parametric distributions as needed for a background model, but may adapt too quickly to the foreground objects.

A persistent particle filter is proposed, following bacterial persistence. Bacterial persistence is linked to the random switch of bacteria between two states: A normal growing cell and a dormant but persistent cell. The dormant cells can survive stress such as antibiotics. When a dormant cell switches to a normal status after the stress is over, bacterial growth continues.

Similar to bacteria, particles will switch between dormant and active states, where dormant particles will not adapt to the changing environment. A further modification of particle filters allows discontinuous jumps into new parameters enabling foreground objects to join the background when they stop moving. This can also quickly build multi-modal distributions.

Index Terms— Object detection and tracking, Tracking filters, Background model, Particle filter

1. INTRODUCTION

Background subtraction is a common approach to extract moving objects from a video sequence, and is a fundamental step in many video surveillance applications. Each pixel in a new frame is compared against a background model, and is recognized as a foreground object if it differs significantly from the background model.

Simple background models such as [1] assume that the color at each pixel over time can be modeled by a unimodal distribution. However, this assumption fails when the scene contains background motion, such as trees moving in the wind or ripples on water, and when the camera is shaking.

Non stationary backgrounds have been estimated by the Mixture of Gaussians (MoG) technique [2, 3, 4]. The Gaus-

sians' parameters are modified at each time step to reflect the changes in the scene. If the adaptation of the Gaussian parameters to the changes is fast, slow objects can be absorbed into the background model. When the adaptation is slow the model has trouble detecting fast changes to the background such as those produced by illumination change [5].

Modeling the background distribution with histograms is a simple, yet robust, technique [6]. Pixel histograms are non parametric and can account for any type of multi-modal distribution. However, the number of bins l in the histogram is fixed and this value can greatly influence the results depending on the video being processed. Moreover, this technique does not scale gracefully. To model color videos the histogram will contain l^3 bins in RGB space, which produces histograms that are large and sparse. Addressing this problem by decoupling the channels and using 3 separate histograms reduces the memory required to $3l$ bins but can cause classification errors.

In [7] Kim *et al.* outline a non parametric, compressed representation of the distribution using a codebook model. The model is built using a long training period where it is assumed that no objects are in the scene.

A Persistent Particle Filter has the ability to model arbitrary multi-modal distributions, and thus can handle repetitive background movements and illumination changes. Memory needs scale easily to 3D color space, without making independence assumptions that sacrifice accuracy, and no training period without moving objects is needed.

2. BACKGROUND MODELS BY PARTICLE FILTERS

In this section we describe a naive use of Particle Filters for background modeling. In Sec. 3 we analyze its deficiencies and describe improvements that make Particle Filters more useful for background modeling. As space precludes a detailed description of Particle Filters we refer the reader to [8] for an in depth review.

Particle Filtering is a technique for estimating the hidden state X_t of a dynamical system at each time t , conditioned on sensor measurements. The goal of the Particle Filter is to

estimate the posterior probability density over the state space by using a set of samples (particles) of the state x_t^i . Each such sample is associated with a weight w_t^i that indicates the quality, or importance, of that sample. The set of particles is described as

$$\mathcal{X}_t = \{ \langle x_t^i, w_t^i \rangle \mid i = 1, \dots, N \}. \quad (1)$$

Background modeling uses one particle filter per pixel, estimating the color distribution of each pixel without making parametric assumptions. The state of pixel r at time t is the set $\mathcal{X}_t(r)$ where particles are vectors over the RGB space.

Particle Filter's three recursive stages can be implemented as follows for background subtraction:

Prediction: For a static camera and a relatively static background, we define a simple transition function

$$\hat{x}_t^i(r) = x_{t-1}^i(r) + \mathcal{N}(0, \sigma), \quad (2)$$

where \mathcal{N} is sampled from a Normal distribution with mean zero and standard deviation σ . This allows the model to adapt to slow background changes, and adds robustness for poor video quality.

Update: In the update stage, the weight of the i 'th particle of pixel r , $x_t^i(r)$, is set according to the particle's distance to the current pixel value:

$$w_{t+1}^i(r) = \exp\left(\frac{-\|\hat{x}_{t+1}^i(r) - I_{t+1}(r)\|_\infty^2}{2\sigma_t^2(r)}\right), \quad (3)$$

where the norm $\|\cdot\|_\infty$ uses the largest difference among the color channels. $I_t(r)$ is the value of pixel r in frame t , and $\sigma_t(r)$ is the standard deviation of the pixel, calculated by an exponentially decaying window:

$$\mu_t(r) = \alpha \cdot \mu_{t-1}(r) + (1 - \alpha) \cdot I_t(r) \quad (4)$$

$$\sigma_t^2(r) = \alpha \cdot \sigma_{t-1}^2(r) + (1 - \alpha) \cdot (I_t(r) - \mu_t(r))^2 \quad (5)$$

In all our experiments we used $\alpha = 0.99$.

Re-sample: Re-sampling is done using a low variance sampler [8]. The low variance sampler uses N equally spread samples whose phase is determined by a single random draw.

2.1. Probability map and segmentation

The background probability of each pixel can be computed from its particle set using an average of the strongest particles. First, the particles are sorted according to their weights in descending order. Averaging the best particles is done by

$$P_b(r) = \frac{1}{K} \sum_{i=1}^K w^i(r) \quad (6)$$

Where $w^i(r)$ are the sorted particles with weights as defined in Eq. (3), and $K = 0.75N$.

Foreground segmentation is achieved by thresholding the probability map P_b by a predefined value P

$$\text{Foreground}(r) = \begin{cases} 0 & P_b(r) < P \\ 1 & \text{otherwise} \end{cases}. \quad (7)$$

We used $P = 0.3$ in all our experiments.

3. PERSISTENT PARTICLES

Particle Filters usually require a large number of particles in order to accurately model complex distributions. For a background modeling algorithm to be practical, the amount of memory used should be minimal, and thus we need to limit the size of the particle set. In the following we outline how to overcome the difficulties introduced by using a small amount of particles.

3.1. Dormant Particles

A Typical scene for surveillance cameras can involve a number of challenges that any background subtraction algorithm needs to cope with. These include dynamic background areas such as moving trees, shaking cameras, and foreground objects that may fuse into the background, like a car pulling into a parking space and staying there. The algorithm should be able to differentiate the latter from another scenario in which the car only pauses for a while (for instance while waiting for a traffic light) and then moves on. These types of scenarios emphasize the multi-modal nature of the background scene. While Particle Filters can represent arbitrary distributions this can require a large number of samples. When using only a small particle set, the re-sampling stage can cause one good particle (a particle whose color is close to the current color of the pixel) to drastically change the entire set, collapsing it into a unimodal representation. This is known as the Loss of Diversity problem. To address these issues we introduce *Persistent Particles*, particles which keep their estimation even when the rest of the sample population has shifted to a different region of the state space.

This is inspired by the behavior of microbial population. Bacteria switch randomly between two states: (i) a state of normal growth; (ii) a dormant state with reduced vulnerability to stress (e.g. antibiotics), enabling survival under stress condition where active bacteria do not survive [9].

We propose to divide the particles into two subsets: *Active* particles and dormant or *Persistent* particles. Particles switch from active to persistent and vice versa when certain conditions are met. A particle is considered active if its current estimate of the background color (its RGB-vector) is close to the current pixel value. Otherwise it is considered persistent.

$$\text{Persist}(x(r)_t^i) = \begin{cases} 1 & \exp\left(\frac{-\|x_t^i(r) - I_t(r)\|_\infty^2}{2\sigma_t^2(r)}\right) < P \\ 0 & \text{otherwise} \end{cases}, \quad (8)$$

where P is the probability threshold as used in Eq. (7).

Active particles behave as described in Sec. 2. Persistent particles have a different prediction step, and don't participate in the re-sampling stage. For a persistent particle $x(r)_t^i$ the prediction process is

$$\hat{x}(r)_{t+1}^i = \begin{cases} I(r)_{t+1} & p < T_1 \\ x(r)_t^i & \text{else} \end{cases}, \quad (9)$$

where p is a uniform random number in $[0, 1]$ and T_1 is the persistence threshold which indicates the duration in which particles remain in the dormant state. In our experiments we used $T_1 = 1/1000$, implying that each particle has a 1/1000 probability to switch to the observed color.

Examining the parking car example we can now understand the effect of these persistent particles. When a car enters the scene, the particles, tracking the background color, will switch into the dormant phase and thus will not track the color of the car even when it stops at the traffic light. When the car parks, the particles will gradually switch into active state and will start following its color distribution and eventually merge it into the background. The amount of time for a stopped foreground object to merge into the background is determined by the value of T_1 .

3.2. Color Jump For Scene Change

While the persistent particles ensure the multi-modal behavior of the sample population, it can take a significant amount of time, in the order of $1/T_1$ for the population to achieve a good representation of the pixel's histogram. When faced with a continuous, high frequency, change in the background, such as moving trees, we would like the filter to adapt quickly to the dynamic nature of the pixel. To handle such cases we give a pixel the ability to switch one particle to the changing scene. But such changes must be separated in time by enough frames. A particle (either active or dormant) is selected randomly and its value is switched to the current color if

$$\|I_t(r) - I_{t-1}(r)\|_\infty > T_2, \quad (10)$$

where in our experiments we use $T_2 = 30$. At least 10 frames should separate two such switches. This allows for quick adaptation of the filter for dynamic textures without contaminating the particle set by passing objects. Due to the particles' jump ability, the particle set can adapt quickly to such changes and hold the various colors that repeat in the background. Fig. 1 shows a scene in which trees are moving in the wind. It shows how the particle set successfully reproduce the background distribution.

It should be noted that in this section we address significant changes between successive frames, while in the handling of the dormant particles we examine differences between a particle and the latest frame.

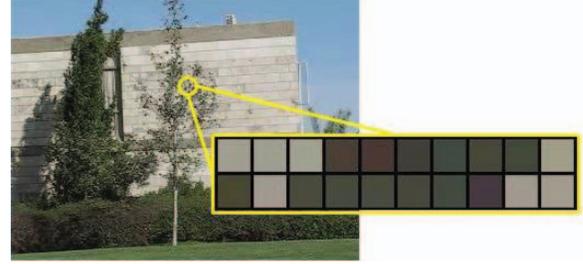


Fig. 1. Estimation of background distribution in dynamic backgrounds, such as trees moving in the wind. Using the particle's jump ability, the particle set can quickly represent the various background colors. The pixel marked by a circle displays both wall and tree colors, as captured by its particles.

4. EXPERIMENTS

Our algorithm was tested on numerous video sequences. In all our tests we assigned $T_1 = 0.001$ (Eq. 9), $T_2 = 30$ (Eq. 10), $P = 0.3$ (Eq. 7), and $\alpha = 0.99$ (Eq. 4). The particle sets contained 20 particles, which were initialized by the result of averaging the first 100 frames of the sequence. We did not require that these frames be without moving objects. For color videos of size 320×240 it runs at approximately 8 fps.

We compare our results with the MoG algorithm [2] and the codebook technique [7], both as implemented in OpenCV [10]. For the Codebook algorithm we used a 100 frame training period. For the MoG we used 5 Gaussian kernels. We also implemented the histograms method [6]. To reduce computation we used three color histograms quantified to 32 colors in each channel, and did not use the spatial neighbors of a pixel. We used an α value of 0.999 for the histogram method. This is equivalent to our T_1 value which controls the speed at which foreground objects can be assimilated into the background model.

In our comparisons we did not use any post processing techniques, such as morphological operations, in order to make it easier to judge the capabilities of each algorithm. Fig. 3 shows frames from a number of different video sequences, and the results of the four algorithms on them¹. Bacteria-Filters seems to perform best, and the histograms method second best.

Foreground objects that stop and later on resume motion can pose a problem for many background modeling algorithms. Ideally the algorithm should retain knowledge of the background colors that are occluded by the object so that when the object continues its movement it won't miss classify the background as a new object. If however the foreground object remains unmoving eventually the algorithm should assimilate it into the background model. Fig. 2 illustrates how

¹The authors thank the ViSOR repository, and Institut für Algorithmen und Kognitive Systeme for providing some of the videos presented.

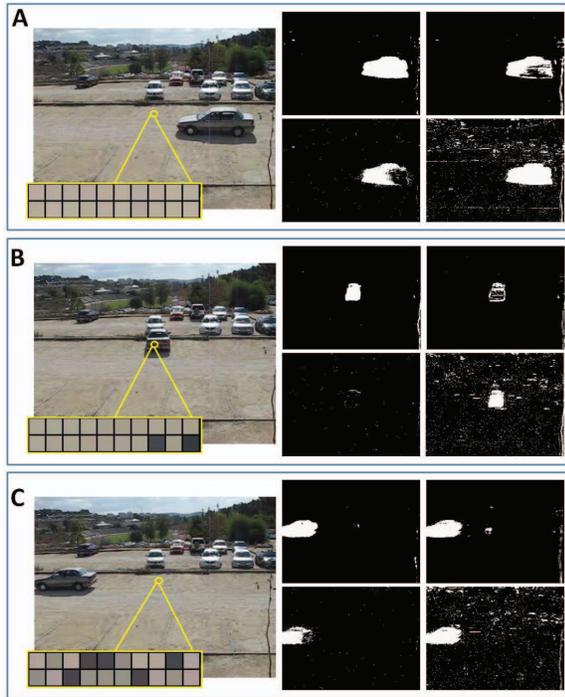


Fig. 2. The particle set gradually adjusts to the new colors of a stopped foreground object allowing accurate segmentation. Results are ordered left to right, top to bottom: Bacteria-Filters, Histograms, Mixture of Gaussians, Codebook.

the particles' ability to switch into the persistent form solves this problem. It shows a car pulling into a parking space (Fig 2A), wait for a while (Fig 2B), and pull out (Fig 2C). As can be shown, the particle set gradually integrates the car's colors into the background model. When the car exits, the set holds information on both the car and the road, and so is able to correctly segment the scene. As the sequence continues more particles shift into tracking the car colors, and so if it had stayed it would have been merged into the background.

5. DISCUSSION

In this paper, a novel background modeling technique has been introduced. Our algorithm is based on the well known Particle Filter framework which has been shown to be a robust and efficient technique for many applications. We have modified this framework, allowing particles to become persistent, or dormant, and thus avoid being replaced due to a few outlier measurements, and a jump ability which enables particles to adapt quickly to multi modal distributions. It was shown that this technique can accurately model the background color distribution even when faced with repetitive background movement, multi modal backgrounds and illumination changes.

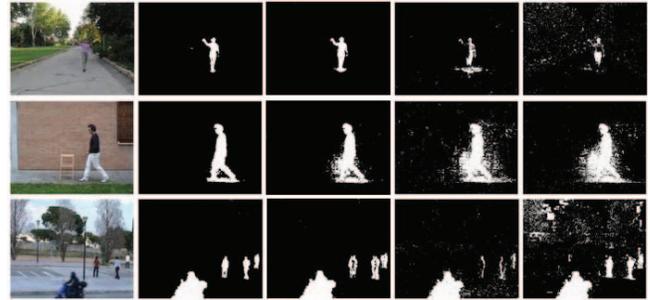


Fig. 3. Examples of foreground segmentation. From left to right: Input frame, Bacteria-Filters, Histograms, Mixture of Gaussians, Codebook.

References

- [1] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE-T-PAMI*, vol. 19, pp. 780–785, 1997.
- [2] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *European Workshop Advanced Video Based Surveillance Systems*, 2001.
- [3] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR'99*, 1999, vol. 2, pp. 246–252.
- [4] D.S. Lee, J.J. Hull, and B. Erol, "A bayesian framework for gaussian mixture background modeling," in *ICIP'03*, 2003, vol. 3, pp. 973–976.
- [5] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *ICCV'99*, 1999, vol. 1, pp. 255–261.
- [6] T. Ko, S. Soatto, and D. Estrin, "Background subtraction on distributions," in *ECCV'08*, 2008, pp. 276–289.
- [7] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.
- [8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, 2005.
- [9] N.Q. Balaban, J. Merrin, R. Chait, L. Kowalik, and S. Leibler, "Bacterial persistence as a phenotypic switch," *Science*, vol. 305, pp. 1622–1625, 2004.
- [10] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, Cambridge, MA, 2008.