Adaptive background mixture models for real-time tracking

Chris Stauffer W.E.L Grimson

The Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

A common method for real-time segmentation of moving regions in image sequences involves "background subtraction," or thresholding the error between an estimate of the image without moving objects and the current image. The numerous approaches to this problem differ in the type of background model used and the procedure used to update the model. This paper discusses modeling each pixel as a mixture of Gaussians and using an on-line approximation to update the model. The Gaussian distributions of the adaptive mixture model are then evaluated to determine which are most likely to result from a background process. Each pixel is classified based on whether the Gaussian distribution which represents it most effectively is considered part of the background model.

This results in a stable, real-time outdoor tracker which reliably deals with lighting changes, repetitive motions from clutter, and long-term scene changes. This system has been run almost continuously for 16 months, 24 hours a day, through rain and snow.

1 Introduction

In the past, computational barriers have limited the complexity of real-time video processing applications. As a consequence, most systems were either too slow to be practical, or succeeded by restricting themselves to very controlled situations. Recently, faster computers have enabled researchers to consider more complex, robust models for real-time analysis of streaming data. These new methods allow researchers to begin modeling real world processes under varying conditions.

Consider the problem of video surveillance and monitoring. A robust system should not depend on careful placement of cameras. It should also be robust to whatever is in its visual field or whatever lighting effects occur. It should be capable of dealing with movement through cluttered areas, objects overlapping in the visual field, shadows, lighting changes, effects of moving elements of the scene (e.g. swaying trees), slow-moving objects, and objects being introduced or removed from the scene. Traditional approaches based on backgrounding methods typically fail in these general situations. Our goal is to create a robust, adaptive tracking system that is flexible enough to handle variations in lighting, moving scene clutter, multiple moving objects and other arbitrary changes to the observed scene. The resulting tracker is primarily geared towards scene-level video surveillance applications.

1.1 Previous work and current shortcomings

Most researchers have abandoned non-adaptive methods of backgrounding because of the need for manual initialization. Without re-initialization, errors in the background accumulate over time, making this method useful only in highly-supervised, short-term tracking applications without significant changes in the scene.

A standard method of adaptive backgrounding is averaging the images over time, creating a background approximation which is similar to the current static scene except where motion occurs. While this is effective in situations where objects move continuously and the background is visible a significant portion of the time, it is not robust to scenes with many moving objects particularly if they move slowly. It also cannot handle bimodal backgrounds, recovers slowly when the background is uncovered, and has a single, predetermined threshold for the entire scene.

Changes in scene lighting can cause problems for many backgrounding methods. Ridder et al.[5] modeled each pixel with a Kalman Filter which made their system more robust to lighting changes in the scene. While this method does have a pixel-wise automatic threshold, it still recovers slowly and does not handle bimodal backgrounds well. Koller et al.[4] have successfully integrated this method in an automatic traffic monitoring application.

Pfinder[7] uses a multi-class statistical model for the tracked objects, but the background model is a single Gaussian per pixel. After an initialization period where the room is empty, the system reports good results. There have been no reports on the success of this tracker in outdoor scenes.

Friedman and Russell[2] have recently implemented a pixel-wise EM framework for detection of vehicles that bears the most similarity to our work. Their method attempts to explicitly classify the pixel values into three separate, predetermined distributions corresponding to the road color, the shadow color, and colors corresponding to vehicles. Their attempt to mediate the effect of shadows appears to be somewhat successful, but it is not clear what behavior their system would exhibit for pixels which did not contain these three distributions. For example, pixels may present a single background color or multiple background colors resulting from repetitive motions, shadows, or reflectances.

1.2 Our approach

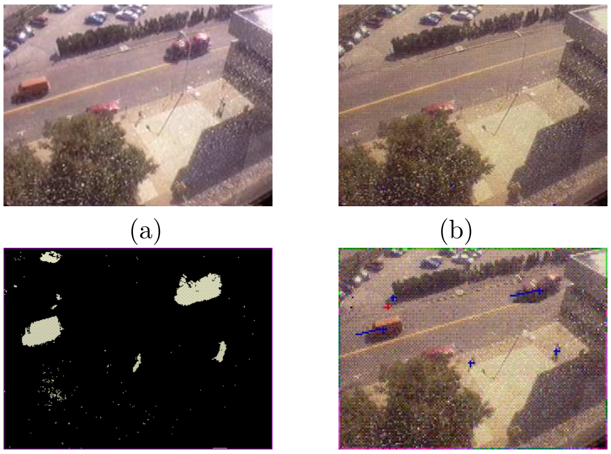
Rather than explicitly modeling the values of all the pixels as one particular type of distribution, we simply model the values of a particular pixel as a mixture of Gaussians. Based on the persistence and the variance of each of the Gaussians of the mixture, we determine which Gaussians may correspond to background colors. Pixel values that do not fit the background distributions are considered foreground until there is a Gaussian that includes them with sufficient, consistent evidence supporting it.

Our system adapts to deal robustly with lighting changes, repetitive motions of scene elements, tracking through cluttered regions, slow-moving objects, and introducing or removing objects from the scene. Slowly moving objects take longer to be incorporated into the background, because their color has a larger variance than the background. Also, repetitive variations are learned, and a model for the background distribution is generally maintained even if it is temporarily replaced by another distribution which leads to faster recovery when objects are removed.

Our backgrounding method contains two significant parameters – α, the learning constant and T, the proportion of the data that should be accounted for by the background. Without needing to alter parameters, our system has been used in an indoors, human-computer interface application and, for the past 16 months, has been continuously monitoring outdoor scenes.

2 The method

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. If only lighting changed over time, a single, adaptive Gaussian per pixel would be sufficient. In practice, multiple surfaces often appear in the view frustum of a particular pixel and the lighting conditions change. Thus, multiple, adaptive Gaussians are necessary. We use a mixture of adaptive Gaussians to approximate this process.



(a) (b) (c) (d)

Figure 1: *The execution of the program. (a) the current image, (b) an image composed of the means of the most probable Gaussians in the background model, (c) the foreground pixels, (d) the current image with tracking information superimposed. Note: while the shadows are foreground in this case,if the surface was covered by shadows a significant amount of the time, a Gaussian representing those pixel values may be significant enough to be considered background.*

Each time the parameters of the Gaussians are updated, the Gaussians are evaluated using a simple heuristic to hypothesize which are most likely to be part of the "background process." Pixel values that do not match one of the pixel's "background" Gaussians are grouped using connected components. Finally, the connected components are tracked from frame to frame using a multiple hypothesis tracker. The process is illustrated in Figure 1.

2.1 Online mixture model

We consider the values of a particular pixel over time as a "pixel process". The "pixel process" is a time series of pixel values, e.g. scalars for grayvalues or vectors for color images. At any time, t, what is known about a particular pixel, $\{x_0, y_0\}$, is its history

$$\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\} \tag{1}$$

where I is the image sequence. Some "pixel processes" are shown by the (R,G) scatter plots in Figure 2(a)-(c)

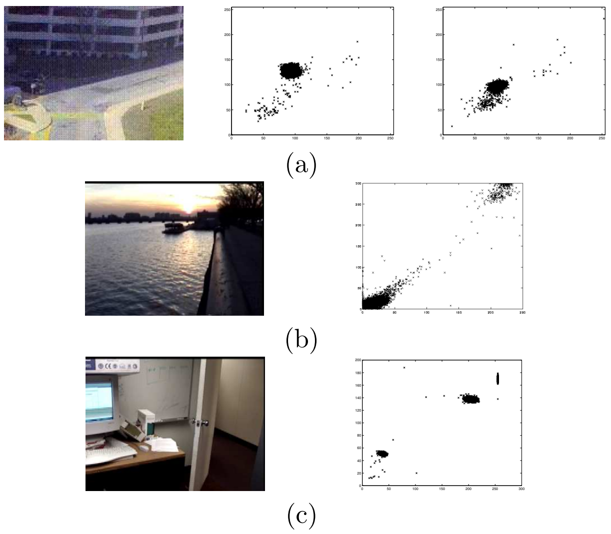


Figure 2: *This figure contains images and scatter plots of the red and green values of a single pixel from the image over time. It illustrates some of the difficulties involved in real environments. (a) shows two scatter plots from the same pixel taken 2 minutes apart. This would require two thresholds. (b) shows a bi-model distribution of a pixel values resulting from specularities on the surface of water. (c) shows another bi-modality resulting from monitor flicker.*

which illustrate the need for adaptive systems with automatic thresholds. Figure 2(b) and (c) also highlight a need for a multi-modal representation.

The value of each pixel represents a measurement of the radiance in the direction of the sensor of the first object intersected by the pixel's optical ray. With a static background and static lighting, that value would be relatively constant. If we assume that independent, Gaussian noise is incurred in the sampling process, its density could be described by a single Gaussian distribution centered at the mean pixel value. Unfortunately, the most interesting video sequences involve lighting changes, scene changes, and moving objects.

If lighting changes occurred in a static scene, it would be necessary for the Gaussian to track those changes. If a static object was added to the scene and was not incorporated into the background until it had been there longer than the previous object, the corresponding pixels could be considered foreground for arbitrarily long periods. This would lead to accumulated errors in the foreground estimation, resulting in poor tracking behavior. These factors suggest that more recent observations may be more important in determining the Gaussian parameter estimates.

An additional aspect of variation occurs if moving objects are present in the scene. Even a relatively consistently colored moving object is generally expected to produce more variance than a "static" object. Also, in general, there should be more data supporting the background distributions because they are repeated, whereas pixel values for different objects are often not the same color.

These are the guiding factors in our choice of model and update procedure. The recent history of each pixel, $\{X_1, ..., X_t\}$, is modeled by a mixture of K Gaussian distributions. The probability of observing the current pixel value is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2}$$

where K is the number of distributions, $\omega_{i,t}$ is an estimate of the weight (what portion of the data is accounted for by this Gaussian) of the i^{th} Gaussian in the mixture at time t, $\mu_{i,t}$ is the mean value of the i^{th} Gaussian in the mixture at time t, $\Sigma_{i,t}$ is the covariance matrix of the i^{th} Gaussian in the mixture at time t, and where η is a Gaussian probability density function

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)} \tag{3}$$

K is determined by the available memory and computational power. Currently, from 3 to 5 are used. Also, for computational reasons, the covariance matrix is assumed to be of the form:

$$\Sigma_{k,t} = \sigma_k^2 \mathbf{I} \tag{4}$$

This assumes that the red, green, and blue pixel values are independent and have the same variances. While this is certainly not the case, the assumption allows us to avoid a costly matrix inversion at the expense of some accuracy.

Thus, the distribution of recently observed values of each pixel in the scene is characterized by a mixture of Gaussians. A new pixel value will, in general, be represented by one of the major components of the mixture model and used to update the model.

If the pixel process could be considered a stationary process, a standard method for maximizing the likelihood of the observed data is *expectation maximization*[1]. Unfortunately, each pixel process varies over time as the state of the world changes, so we use an approximate method which essentially treats each new observation as a sample set of size 1 and uses standard learning rules to integrate the new data.

Because there is a mixture model for every pixel in the image, implementing an exact EM algorithm on a window of recent data would be costly. Instead, we implement an on-line K-means approximation. Every new pixel value, X_t, is checked against the existing K Gaussian distributions, until a match is found. A match is defined as a pixel value within 2.5 standard deviations of a distribution[1]. This threshold can be perturbed with little effect on performance. This is effectively a per pixel/per distribution threshold. This is extremely useful when different regions have different lighting (see Figure 2(a)), because objects which appear in shaded regions do not generally exhibit as much noise as objects in lighted regions. A uniform threshold often results in objects disappearing when they enter shaded regions.

If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and low prior weight.

The prior weights of the K distributions at time t, ω_k, t, are adjusted as follows

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{5}$$

where α is the learning rate[2] and $M_{k,t}$ is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized. $1/\alpha$ defines the time constant which determines the speed at which the distribution's parameters change. $\omega_{k,t}$ is effectively a causal low-pass filtered average of the (thresholded) posterior probability that pixel values have matched model k given observations from time 1 through t. This is equivalent to the expectation of this value with an exponential window on the past values.

The μ and σ parameters for unmatched distributions remain the same. The parameters of the distribution which matches the new observation are updated as follows

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t \tag{6}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{7}$$

where

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \tag{8}$$

which is effectively the same type of causal low-pass filter as mentioned above, except that only the data which matches the model is included in the estimation.

One of the significant advantages of this method is that when something is allowed to become part of the background, it doesn't destroy the existing model of the background. The original background color remains in the mixture until it becomes the K^{th} most probable and a new color is observed. Therefore, if an object is stationary just long enough to become part of the background and then it moves, the distribution describing the previous background still exists with the same μ and σ^2, but a lower ω and will be quickly re-incorporated into the background.

2.2 Background Model Estimation

As the parameters of the mixture model of each pixel change, we would like to determine which of the Gaussians of the mixture are most likely produced by background processes. Heuristically, we are interested in the Gaussian distributions which have the most supporting evidence and the least variance.

To understand this choice, consider the accumulation of supporting evidence and the relatively low variance for the "background" distributions when a static, persistent object is visible. In contrast, when a new object occludes the background object, it will not, in general, match one of the existing distributions which will result in either the creation of a new distribution or the increase in the variance of an existing distribution. Also, the variance of the moving object is expected to remain larger than a background pixel until the moving object stops. To model this, we need a method for deciding what portion of the mixture model best represents background processes.

First, the Gaussians are ordered by the value of ω/σ. This value increases both as a distribution gains more evidence and as the variance decreases. After re-estimating the parameters of the mixture, it is sufficient to sort from the matched distribution towards the most probable background distribution, because only the matched models relative value will have changed. This ordering of the model is effectively an ordered, open-ended list, where the most likely background distributions remain on top and the less probable transient background distributions gravitate towards the bottom and are eventually replaced by new distributions.

Then the first B distributions are chosen as the background model, where

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > T \right) \tag{9}$$

[1] Depending on the curtosis of the noise, some percentage of the data points "generated" by a Gaussian will not "match". The resulting random noise is easily ignored by neglecting connected components containing only 1 or 2 pixels.

[2] While this rule is easily interpreted an an interpolation between two points, it is often shown in the equivalent form: $\omega_{k,t} = \omega_{k,t-1} + \alpha(M_{k,t} - \omega_{k,t-1})$

where T is a measure of the minimum portion of the data that should be accounted for by the background. This takes the "best" distributions until a certain portion, T, of the recent data has been accounted for. If a small value for T is chosen, the background model is usually unimodal. If this is the case, using only the most probable distribution will save processing.

If T is higher, a multi-modal distribution caused by a repetitive background motion (e.g. leaves on a tree, a flag in the wind, a construction flasher, etc.) could result in more than one color being included in the background model. This results in a transparency effect which allows the background to accept two or more separate colors.

2.3 Connected components

The method described above allows us to identify foreground pixels in each new frame while updating the description of each pixel's process. These labeled foreground pixels can then be segmented into regions by a two-pass, connected components algorithm [3].

Because this procedure is effective in determining the whole moving object, moving regions can be characterized not only by their position, but size, moments, and other shape information. Not only can these characteristics be useful for later processing and classification, but they can aid in the tracking process.

2.4 Multiple Hypothesis Tracking

While this section is not essential in the understanding of the background subtraction method, it will allow one to better understand and evaluate the results in the following sections.

Establishing correspondence of connected components between frames is accomplished using a linearly predictive multiple hypotheses tracking algorithm which incorporates both position and size. We have implemented an online method for seeding and maintaining sets of Kalman filters.

At each frame, we have an available pool of Kalman models and a new available pool of connected components that they could explain. First, the models are probabilistically matched to the connected regions that they could explain. Second, the connected regions which could not be sufficiently explained are checked to find new Kalman models. Finally, models whose fitness (as determined by the inverse of the variance of its prediction error) falls below a threshold are removed.

Matching the models to the connected components involves checking each existing model against the available pool of connected components which are larger than a pixel or two. All matches are used to update the corresponding model. If the updated model has sufficient fitness, it will be used in the following frame. If no match is found a "null" match can be hypothesized which propogates the model as expected and decreases its fitness by a constant factor.

The unmatched models from the current frame and the previous two frames are then used to hypothesize new models. Using pairs of unmatched connected components from the previous two frames, a model is hypothesized. If the current frame contains a match with sufficient fitness, the updated model is added to the existing models. To avoid possible combinatorial explosions in noisy situations, it may be desirable to limit the maximum number of existing models by removing the least probable models when excessive models exist. In noisy situations (e.g. ccd cameras in low-light conditions), it is often useful to remove the short tracks that may result from random correspondances. Further details of this method can be found at http://www.ai.mit.edu/projects/vsam/.

3 Results

On an SGI O2 with a R10000 processor, this method can process 11 to 13 frames a second (frame size 160x120pixels). The variation in the frame rate is due to variation in the amount of foreground present. Our tracking system has been effectively storing tracking information for five scenes for over 16 months[6]. Figure 3 shows accumulated tracks in one scene over the period of a day.

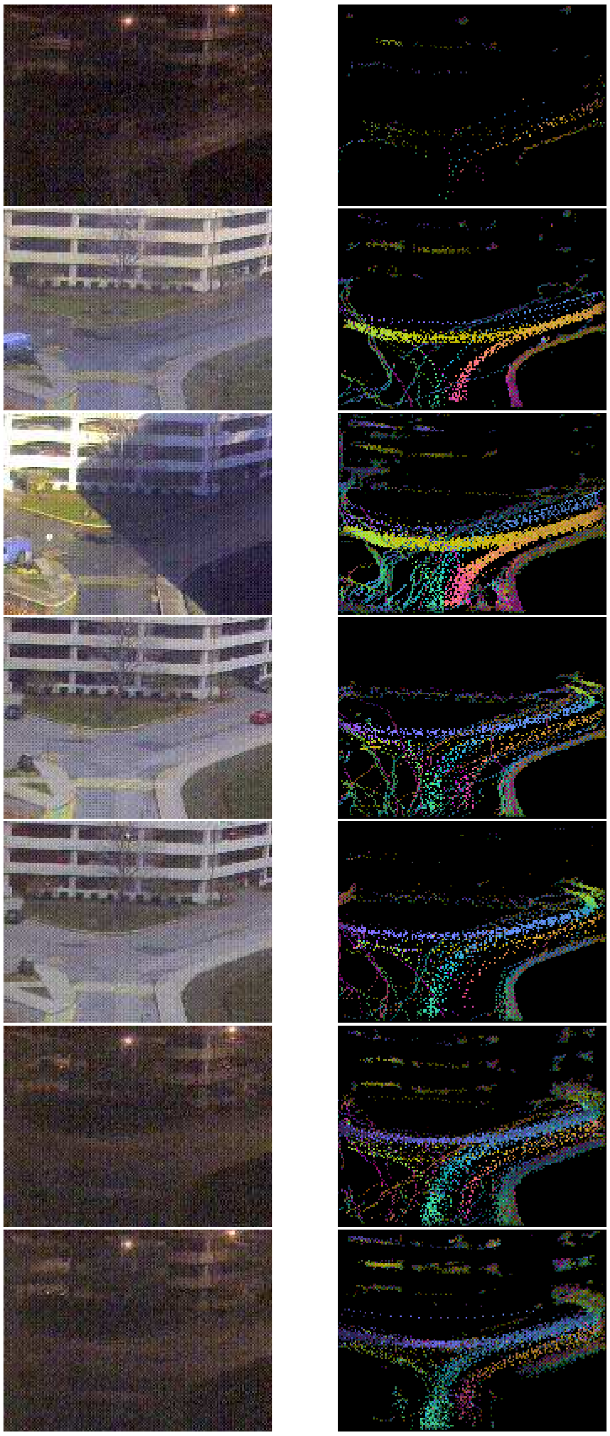
While quick changes in cloud cover (relative to α, the learning rate) can sometimes necessitate a new set of background distributions, it will stabilize within 10-20 seconds and tracking will continue unhindered.

Because of the stability and completeness of the representation it is possible to do some simple classification. Figure 4 shows the classification of objects which appeared in a scene over a 10 minute period using a simple binary threshold on the time-averaged aspect ratio of the object. Tracks lasting less than a second were removed.

Every object which entered this scene – in total, 33 cars and 34 people – was tracked. It successfully classified every car except in one case, where it classified two cars as the same object because one car entered the scene simultaneously with another car leaving at the same point. It found only one person in two cases where two people where walking in physical contact. It also double counted 2 objects because their tracks were not matched properly.

4 Applicability

When deciding on a tracker to implement, the most important information to a researcher is where the



(a) (b)

Figure 3: *This figure shows consecutive hours of tracking from 6am to 9am and 3pm to 7pm. (a) shows the image at the time the template was stored and (b) show the accumulated tracks of the objects over that time. Color encodes direction and intensity encodes size. The consistency of the colors within particular regions reflects the consistency of the speed, direction, and size parameters which have been acquired.*

(a)

(b)

Figure 4: *This figure shows which objects in the scene were classified as people or cars using simple heuristics on the aspect ratio of the observed object. Its accuracy reflects the consistency of the connected regions which are being tracked.*

tracker is applicable. This section will endeavor to pass on some of the knowledge we have gained through our experiences with this tracker.

The tracking system has the most difficulty with scenes containing high occurrences of objects that visually overlap. The multiple hypothesis tracker is not extremely sophisticated about reliably dissambiguating objects which cross. This problem can be compounded by long shadows, but for our applications it was much more desirable to track an object and its shadow and avoid cropping or missing dark objects than it was to attempt to remove shadows. In our experience, on bright days when the shadows are the most significant, both shadowed regions and shady sides of dark objects are black (not dark green, not dark red, etc.).

The good news is that the tracker was relatively robust to all but relatively fast lighting changes (e.g. flood lights turning on and partly cloudy, windy days). It successfully tracked outdoor scenes in rain, snow, sleet, hail, overcast, and sunny days. It has also been used to track birds at a feeder, mice at night using Sony NightShot, fish in a tank, people entering a lab, and objects in outdoor scenes. In these environments, it reduces the impact of repetative motions from swaying branches, rippling water, specularities, slow moving objects, and camera and acquisition noise. The system has proven robust to day/night cycles and long-term scene changes. More recent results and project updates are available at http://www.ai.mit.edu/projects/vsam/.

5 Future work

As computers improve and parallel architectures are investigated, this algorithm can be run faster, on larger images, and using a larger number of Gaussians in the mixture model. All of these factors will increase performance. A full covariance matrix would further improve performance. Adding prediction to each Gaussian(e.g. the Kalman filter approach), may also lead to more robust tracking of lighting changes.

Beyond these obvious improvements, we are investigating modeling some of the inter-dependencies of the pixel processes. Relative values of neighboring pixels and correlations with neighboring pixel's distributions may be useful in this regard. This would allow the system to model changes in occluded pixels by observations of some of its neighbors.

Our method has been used on grayscale, RGB, HSV, and local linear filter responses. But this method should be capable of modeling any streamed input source in which our assumptions and heuristics are generally valid. We are investigating use of this method with frame-rate stereo, IR cameras, and including depth as a fourth channel(R,G,B,D). Depth is an example where multi-modal distributions are useful, because while disparity estimates are noisy due to false correspondences, those noisy values are often relatively predictable when they result from false correspondences in the background.

In the past, we were often forced to deal with relatively small amounts of data, but with this system we can collect images of moving objects and tracking data robustly on real-time streaming video for weeks at a time. This ability is allowing us to investigate future directions that were not available to us in the past. We are working on activity classification and object classification using literally millions of examples[6].

6 Conclusions

This paper has shown a novel, probabilistic method for background subtraction. It involves modeling each pixel as a separate mixture model. We implemented a real-time approximate method which is stable and robust. The method requires only two parameters, α and T. These two parameters are robust to different cameras and different scenes.

This method deals with slow lighting changes by slowly adapting the values of the Gaussians. It also deals with multi-modal distributions caused by shadows, specularities, swaying branches, computer monitors, and other troublesome features of the real world which are not often mentioned in computer vision. It recovers quickly when background reappears and has a automatic pixel-wise threshold. All these factors have made this tracker an essential part of our activity and object classification research.

This system has been successfully used to track people in indoor environments, people and cars in outdoor environments, fish in a tank, ants on a floor, and remote control vehicles in a lab setting. All these situations involved different cameras, different lighting, and different objects being tracked. This system achieves our goals of real-time performance over extended periods of time without human intervention.

Acknowledgments

This research is supported in part by a grant from DARPA under contract N00014-97-1-0363 administered by ONR and in part by a grant jointly administered by DARPA and ONR under contract N00014-95-1-0600.

References

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

[2] Nir Friedman and Stuart Russell. "Image segmentation in video sequences: A probabilistic approach," In *Proc. of the Thirteenth Conference on Uncertainty in Artificial Intelligence(UAI)*, Aug. 1-3, 1997.

[3] B. K. P. Horn. *Robot Vision*, pp. 66-69, 299-333. The MIT Press, 1986.

[4] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel. "Towards robust automatic traffic scene analysis in real-time." In *Proc. of the International Conference on Pattern Recognition*, Israel, November 1994.

[5] Christof Ridder, Olaf Munkelt, and Harald Kirchner. "Adaptive Background Estimation and Foreground Detection using Kalman-Filtering," *Proceedings of International Conference on recent Advances in Mechatronics*, ICRAM'95, UNESCO Chair on Mechatronics, 193-199, 1995.

[6] W.E.L. Grimson, Chris Stauffer, Raquel Romano, and Lily Lee. "Using adaptive tracking to classify and monitor activities in a site," In *Computer Vision and Pattern Recognition 1998(CVPR98)*, Santa Barbara, CA. June 1998.

[7] Wren, Christopher R., Ali Azarbayejani, Trevor Darrell, and Alex Pentland. "Pfinder: Real-Time Tracking of the Human Body," In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 1997, vol 19, no 7, pp. 780-785.