Non-parametric Model for Background Subtraction

Ahmed Elgammal, David Harwood, Larry Davis

Computer Vision Laboratory
University of Maryland, College Park, MD 20742, USA
{elgammal,harwood,lsd}@umiacs.umd.edu

Abstract. Background subtraction is a method typically used to segment moving regions in image sequences taken from a static camera by comparing each new frame to a model of the scene background. We present a novel non-parametric background model and a background subtraction approach. The model can handle situations where the background of the scene is cluttered and not completely static but contains small motions such as tree branches and bushes. The model estimates the probability of observing pixel intensity values based on a sample of intensity values for each pixel. The model adapts quickly to changes in the scene which enables very sensitive detection of moving targets. We also show how the model can use color information to suppress detection of shadows. The implementation of the model runs in real-time for both gray level and color imagery. Evaluation shows that this approach achieves very sensitive detection with very low false alarm rates.

Key words: visual motion, active and real time vision, motion detection, non-parametric estimation, visual surveillance, shadow detection

1 Introduction

The detection of unusual motion is the first stage in many automated visual surveillance applications. It is always desirable to achieve very high sensitivity in the detection of moving objects with the lowest possible false alarm rates. Background subtraction is a method typically used to detect unusual motion in the scene by comparing each new frame to a model of the scene background.

If we monitor the intensity value of a pixel over time in a completely static scene (i.e., with no background motion) , then the pixel intensity can be reasonably modeled with a Normal distribution $N(\mu, \sigma^2)$, given the image noise over time can be modeled by a zero mean Normal distribution $N(0, \sigma^2)$. This Normal distribution model for the intensity value of a pixel is the underlying model for many background subtraction techniques. For example, one of the simplest background subtraction techniques is to calculate an average image of the scene with no moving objects, subtract each new frame from this image, and threshold the result.

This basic Normal model can adapt to slow changes in the scene (for example, illumination changes) by recursively updating the model using a simple adaptive filter. This basic adaptive model is used in [1], also Kalman filtering for adaptation is used in [2–4].

In many visual surveillance applications that work with outdoor scenes, the background of the scene contains many non-static objects such as tree branches and bushes whose movement depends on the wind in the scene. This kind of background motion causes the pixel intensity values to vary significantly with time. For example, one pixel can be image of the sky at one frame, tree leaf at another frame, tree branch on a third frame and some mixture subsequently; in each situation the pixel will have a different color.

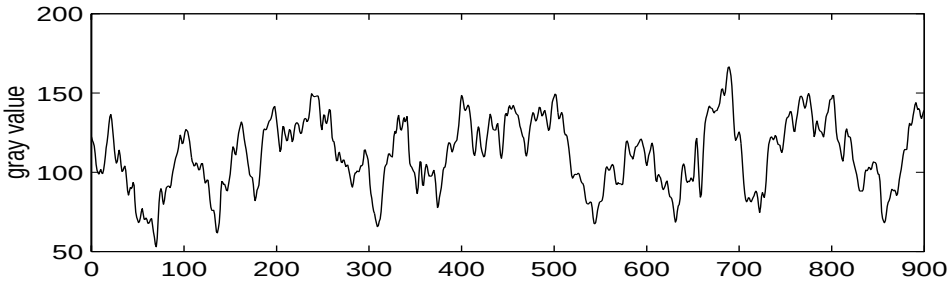


Fig. 1. Intensity value overtime

Fig. 2. Outdoor scene with a circle at the top left corner showing the location of the sample pixel in figure 1

Figure 1 shows how the gray level of a vegetation pixel from an outdoor scene changes over a short period of time (900 frames-30 seconds). The scene is shown at figure 2. Figure 3-a shows the intensity histogram for this pixel. It is clear

that intensity distribution is multi-modal so that the Normal distribution model for the pixel intensity/color would not hold.

In [5] a mixture of three Normal distributions was used to model the pixel value for traffic surveillance applications. The pixel intensity was modeled as a weighted mixture of three Normal distributions: road, shadow and vehicle distribution. An incremental EM algorithm was used to learn and update the parameters of the model. Although, in this case, the pixel intensity is modeled with three distributions, still the uni-modal distribution assumption is used for the scene background, i.e. the road distribution.

In [6, 7] a generalization to the previous approach was presented. The pixel intensity is modeled by a mixture of K Gaussian distributions (K is a small number from 3 to 5) to model variations in the background like tree branch motion and similar small motion in outdoor scenes. The probability that a certain pixel has intensity x_t at time t is estimated as:

$$Pr(x_t) = \sum_{j=1}^{K} \frac{w_j}{(2\pi)^{\frac{d}{2}} \mid \Sigma_j \mid^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t-\mu_j)^T \Sigma_j^{-1}(x_t-\mu_j)} \tag{1}$$

where w_j is the weight, μ_j is the mean and $\Sigma_j = \sigma_j^2 I$ is the covariance for the jth distribution. The K distributions are ordered based on w_j/σ_j^2 and the first B distributions are used as a model of the background of the scene where B is estimated as

$$B = \arg\min_b \left(\frac{\sum_{j=1}^{b} w_j}{\sum_{j=1}^{K} w_j} > T \right) \tag{2}$$

The threshold T is the fraction of the total weight given to the background model. Background subtraction is performed by marking any pixel that is more that 2.5 standard deviations away from any of the B distributions as a foreground pixel. The parameters of the distributions are updated recursively using a learning rate α, where $1/\alpha$ controls the speed at which the model adapts to change.

In the case where the background has very high frequency variations, this model fails to achieve sensitive detection. For example, the 30 second intensity histogram, shown in figure 3-a, shows that the intensity distribution covers a very wide range of gray levels (this would be true for color also.) All these variations occur in a very short period of time (30 seconds.) Modeling the background variations with a small number of Gaussian distribution will not be accurate. Furthermore, the very wide background distribution will result in poor detection because most of the gray level spectrum would be covered by the background model.

Another important factor is how fast the background model adapts to change. Figure 3-b shows 9 histograms of the same pixel obtained by dividing the original time interval into nine equal length subintervals, each contains 100 frames ($3\frac{1}{3}$ seconds.) From these partial histogram we notice that the intensity distribution is changing dramatically over very short periods of time. Using more "short-term" distributions will allow us to obtain better detection sensitivity.

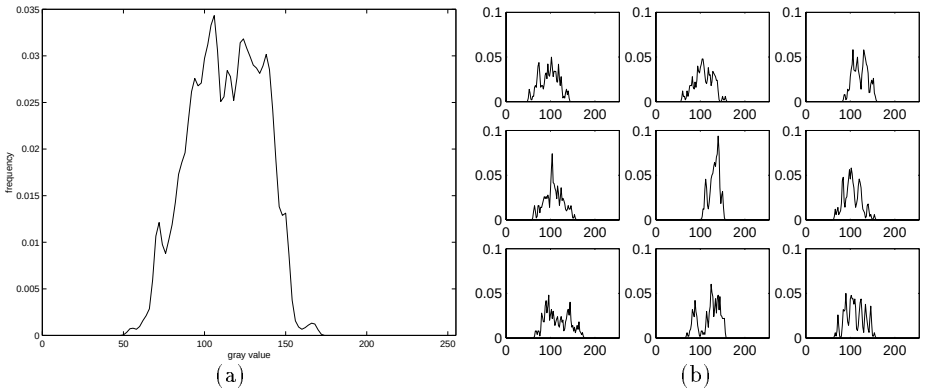


Fig. 3. (a) Histogram of intensity values, (b) Partial histograms

We are faced with the following trade off: if the background model adapts too slowly to changes in the scene, then we will construct a very wide and inaccurate model that will have low detection sensitivity. On the other hand, if the model adapts too quickly, this will lead to two problems: the model may adapt to the targets themselves, as their speed cannot be neglected with respect to the background variations, and it leads to inaccurate estimation of the model parameters.

Our objective is to be able to accurately model the background process non-parametrically. The model should adapt very quickly to changes in the background process, and detect targets with high sensitivity. In the following sections we describe a background model that achieves these objectives. The model keeps a sample for each pixel of the scene and estimates the probability that a newly observed pixel value is from the background. The model estimates these probabilities independently for each new frame. In section 2 we describe the suggested background model and background subtraction process. A second stage of background subtraction is discussed in section 3 that aims to suppress false detections that are due to small motions in the background not captured by the model. Adapting to long-term changes is discussed in section 4. In section 5 we explain how to use color to suppress shadows from being detected.

2 Basic Background Model

2.1 Density Estimation

In this section, we describe the basic background model and the background subtraction process. The objective of the model is to capture very recent information about the image sequence, continuously updating this information to capture fast changes in the scene background. As shown in figure 3-b, the intensity distribution of a pixel can change quickly. So we must estimate the density

function of this distribution at any moment of time given only very recent history information if we hope to obtain sensitive detection.

Let $x_1, x_2, ..., x_N$ be a recent sample of intensity values for a pixel. Using this sample, the probability density function that this pixel will have intensity value x_t at time t can be non-parametrically estimated [8] using the kernel estimator K as

$$Pr(x_t) = \frac{1}{n} \sum_{i=1}^{N} K(x_t - x_i) \tag{3}$$

If we choose our kernel estimator function, K, to be a Normal function $N(0, \Sigma)$, where Σ represents the kernel function bandwidth, then the density can be estimated as

$$Pr(x_t) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{(2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t - x_i)^T \Sigma^{-1} (x_t - x_i)} \tag{4}$$

If we assume independence between the different color channels with a different kernel bandwidths σ_j^2 for the jth color channel, then

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix}$$

and the density estimation is reduced to

$$Pr(x_t) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2}\frac{(x_{tj} - x_{ij})^2}{\sigma_j^2}} \tag{5}$$

Using this probability estimate the, pixel is considered a foreground pixel if $Pr(x_t) < th$ where the threshold th is a global threshold over all the image that can be adjusted to achieve a desired percentage of false positives. Practically, the probability estimation of equation 5 can be calculated in a very fast way using precalculated lookup tables for the kernel function values given the intensity value difference, $(x_t - x_i)$, and the kernel function bandwidth. Moreover, a partial evaluation of the sum in equation 5 is usually sufficient to surpass the threshold at most image pixels, since most of the image is typically sampled from the background. This allows us to construct a very fast implementation of the probability estimation.

Density estimation using a Normal kernel function is a generalization of the Gaussian mixture model, where each single sample of the N samples is considered to be a Gaussian distribution $N(0, \Sigma)$ by itself. This allows us to estimate the density function more accurately and depending only on recent information from the sequence. This also enables the model to quickly "forget" about the past and concentrate more on recent observation. At the same time, we avoid the inevitable errors in parameter estimation, which typically require large amounts of data to be both accurate and unbiased. In section 6.1, we present a comparison

(a) (b)

Fig. 4. Background Subtraction. (a) original image. (b) Estimated probability image.

between the two models. We will show that if both models are given the same amount of memory, and the parameters of the two models are adjusted to achieve the same false positive rates, then the non-parametric model has much higher sensitivity in detection than the mixture of K Gaussians.

Figure 4-b shows the estimated background probability where brighter pixels represent lower background probability pixels.

2.2 Kernel Width Estimation

There are at least two sources of variations in a pixel's intensity value. First, there are large jumps between different intensity values because different objects (sky, branch, leaf and mixtures when an edge passes through the pixel) are projected to the same pixel at different times. Second, for those very short periods of time when the pixel is a projection of the same object, there are local intensity variations due to blurring in the image. The kernel bandwidth, Σ, should reflect the local variance in the pixel intensity due to the local variation from image blur and not the intensity jumps. This local variance will vary over the image and change over time. The local variance is also different among the color channels, requiring different bandwidths for each color channel in the kernel calculation.

To estimate the kernel band width σ_j^2 for the jth color channel for a given pixel we compute the median absolute deviation over the sample for consecutive intensity values of the pixel. That is, the median, m, of $\mid x_i - x_{i+1} \mid$ for each consecutive pair (x_i, x_{i+1}) in the sample, is calculated independently for each color channel. Since we are measuring deviations between two consecutive intensity values, the pair (x_i, x_{i+1}) usually comes from the same local-in-time distribution and only few pairs are expected to come from cross distributions. If we assume that this local-in-time distribution is Normal $N(\mu, \sigma^2)$, then the deviation $(x_i - x_{i+1})$ is Normal $N(0, 2\sigma^2)$. So the standard deviation of the first distribution can be estimated as

$$\sigma = \frac{m}{0.68\sqrt{2}}$$

Since the deviations are integer values, linear interpolation is used to obtain more accurate median values.

3 Suppression of False Detection

In outdoor environments with fluctuating backgrounds, there are two sources of false detections. First, there are false detections due to random noise which should be homogeneous over the entire image. Second, there are false detection due to small movements in the scene background that are not represented in the background model. This can occur, for example, if a tree branch moves further than it did during model generation. Also small camera displacements due to wind load are common in outdoor surveillance and cause many false detections. This kind of false detection is usually spatially clustered in the image and it is not easy to eliminate using morphology or noise filtering because these operations might also affect small and/or occluded targets.

The second stage of detection aim to suppress the false detections due to small and unmodelled movements in the scene background. If some part of the background (a tree branch for example) moves to occupy a new pixel, but it was not part of the model for that pixel, then it will be detected as a foreground object. However, this object will have a high probability to be a part of the background distribution at its original pixel. Assuming that only a small displacement can occur between consecutive frames, we decide if a detected pixel is caused by a background object that has moved by considering the background distributions in a small neighborhood of the detection.

Let x_t be the observed value of a pixel, x, detected as a foreground pixel by the first stage of the background subtraction at time t. We define the pixel displacement probability, $P_{\mathcal{N}}(x_t)$, to be the maximum probability that the observed value, x_t, belongs to the background distribution of some point in the neighborhood $\mathcal{N}(x)$ of x

$$P_{\mathcal{N}}(x_t) = \max_{y \in \mathcal{N}(x)} Pr(x_t \mid B_y)$$

where B_y is the background sample for pixel y and the probability estimation, $Pr(x_t \mid B_y)$, is calculated using the kernel function estimation as in equation 5. By thresholding $P_{\mathcal{N}}$ for detected pixels we can eliminate many false detections due to small motions in the background. Unfortunately, we can also eliminate some true detections by this process, since some true detected pixels might be accidentally similar to the background of some nearby pixel. This happens more often on gray level images. To avoid losing such true detections we add the constraint that the whole detected foreground object must have moved from a nearby location, and not only some of its pixels. We define the component displacement probability, $P_{\mathcal{C}}$, to be the probability that a detected connected component $\mathcal{C}$ has been displaced from a nearby location. This probability is estimated by

$$P_{\mathcal{C}} = \prod_{x \in \mathcal{C}} P_{\mathcal{N}(x)}$$

For a connected component corresponding to a real target, the probability that this component has displaced from the background will be very small. So, a detected pixel x will be considered to be a part of the background only if $(P_{\mathcal{N}}(x) > th_1) \wedge (P_{\mathcal{C}}(x) > th_2)$.

In our implementation, a diameter 5 circular neighborhood is used to determine pixel displacement probabilities for pixels detected from stage one. The threshold th_1 was set to be the same threshold used during the first background subtraction stage which was adjusted to produce a fixed false detection rate. The threshold, th_2, can powerfully discriminate between real moving components and displaced ones since the former have much lower component displacement probabilities.

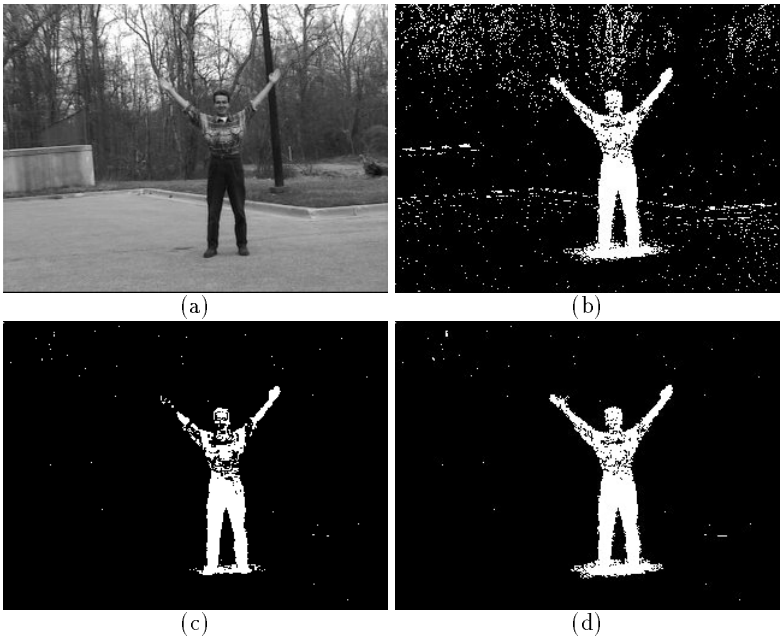


Fig. 5. Effect of the second stage of detection on suppressing false detections

Figure 5 illustrates the effect of the second stage of detection. The result after the first stage is shown in figure 5-b. In this example, the background has not been updated for several seconds and the camera has been slightly displaced during this time interval, so we see many false detection along high contrast edges. Figure 5-c shows the result after suppressing detected pixels with high displacement probability. We eliminates most of the false detections due to displacement, and only random noise that is not correlated with the scene remains as false detections; but some true detected pixel were also lost. The final result of the second stage of the detection is shown in figure 5-d where

the component displacement probability constraint was added. Figure 6-b shows another results where as a result of the wind load the camera is shaking slightly which results in a lot of clustered false detections especially on the edges. After the second stage of detection, figure 6-c, most of these clustered false detection are suppressed while the small target at the left side of the image remains.

(a) (b) (c)

Fig. 6. b) Result after first stage of detection. (c) Result after second stage

4 Updating The Background

In the previous sections it was shown how to detect foreground regions given a recent history sample as a model of the background. This sample contains N intensity values taken over a window in time of size W. The kernel bandwidth estimation requires all the sample to be consecutive in time, i.e., $N = W$ or sample $\frac{N}{2}$ pairs of consecutive intensity values over time W.

This sample needs to be updated continuously to adapt to changes in the scene. The update is performed in a first-in first-out manner. That is, the oldest sample/pair is discarded and a new sample/pair is added to the model. The new sample is chosen randomly from each interval of length $\frac{W}{N}$ frames.

Given a new pixel sample, there are two alternative mechanisms to update the background:

1. Selective Update: add the new sample to the model only if it is classified as a background sample.
2. Blind Update: just add the new sample to the model.

There are tradeoffs to these two approaches. The first enhance detection of the targets, since target pixels are not added to the model. This involves an update decision: we have to decide if each pixel value belongs to the background or not. The simplest way to do this is to use the detection result as an update decision. The problem with this approach is that any incorrect detection decision will result in persistent incorrect detection later, which is a deadlock situations [2]. So for example, if a tree branch might be displaced and stayed fixed in the new location for a long time, it would be continually detected.

The second approach does not suffer from this deadlock situation since it does not involve any update decisions; it allows intensity values that do not belong to the background to be added to the model. This leads to bad detection of the targets (more false negatives) as they erroneously become part of the model. This effect is reduced as we increase the time window over which the sample are taken, as a smaller proportion of target pixels will be included in the sample. But as we increase the time window more false positives will occur because the adaptation to changes is slower and rare events are not as well represented in the sample.

Our objective is to build a background model that adapts quickly to changes in the scene to support sensitive detection and low false positive rates. To achieve this goal we present a way to combine the results of two background models (a long term and a short term) in such a way to achieve better update decisions and avoid the tradeoffs discussed above. The two models are designed to achieve different objectives. First we describe the features of each model.

Short-term model: This is a very recent model of the scene. It adapts to changes quickly to allow very sensitive detection. This model consists of the most recent N background sample values. The sample is updated using a selective-update mechanism, where the update decision is based on a mask $M(p,t)$ where $M(p,t) = 1$ if the pixel p should be updated at time t and 0 otherwise. This mask is driven from the final result of combining the two models.

This model is expected to have two kinds of false positives: false positives due to rare events that are not represented in the model, and persistent false positives that might result from incorrect detection/update decisions due to changes in the scene background.

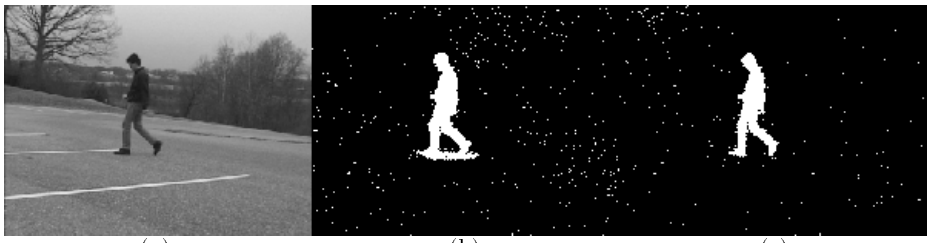
Long-term model: This model captures a more stable representation of the scene background and adapts to changes slowly. This model consists of N sample points taken from a much larger window in time. The sample is updated using a blind-update mechanism, so that every new sample is added to the model regardless of classification decisions. This model is expected to have more false positives because it is not the most recent model of the background, and more false negatives because target pixels might be included in the sample. This model adapts to changes in the scene at a slow rate based on the ratio W/N

Computing the intersection of the two detection results will eliminate the persistence false positives from the short term model and will eliminate as well extra false positives that occur in the long term model results. The only false positives that will remain will be rare events not represented in either model. If this rare event persists over time in the scene then the long term model will adapt to it, and it will be suppressed from the result later.

Taking the intersection will, unfortunately, suppress true positives in the first model result that are false negatives in the second, because the long term model adapts to targets as well if they are stationary or moving slowly. To address this problem, all pixels detected by the short term model that are adjacent to pixels detected by the combination are included in the final result.

5 Shadow detection

The detection of shadows as foreground regions is a source of confusion for subsequent phases of analysis. It is desirable to discriminate between targets and their detected shadows. Color information is useful for suppressing shadows from detection by separating color information from lightness information. Given three color variables, R, G and B, the chromaticity coordinates r, g and b are $r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}$ where $r + g + b = 1$ [9]. Using the chromaticity coordinates in detection has the advantage of being more insensitive to small changes in illumination that are due to shadows. Figure 7 shows the results of detection using both (R, G, B) space and (r, g) space; the figure shows that using the chromaticity coordinates allow detection of the target without detecting their shadows. Notice that the background subtraction technique as described in section 2 can be used with any color space.



(a) (b) (c)

Fig. 7. b) Detection using (R,G,B) color space c) detection using chromaticity coordinates (r,g)

Although using chromaticity coordinates helps suppressing shadows, they have the disadvantage of losing lightness information. Lightness is related to the difference in whiteness, blackness and grayness between different objects [10]. For example, consider the case where the target wears a white shirt and walks against a gray background. In this case there is no color information. Since both white and gray have the same chromaticity coordinates, the target will not be detected.

To address this problem we also need to use a measure of lightness at each pixel. We use $s = R + G + B$ as a lightness measure. Consider the case where the background is completely static, and let the expected value for a pixel be $< r, g, s >$. Assume that this pixel is covered by shadow in frame t and let $< r_t, g_t, s_t >$ be the observed value for this pixel at this frame. Then, it is expected that $\alpha \leq \frac{s_t}{s} \leq 1$. That is, it is expected that the observed value, s_t, will be darker than the normal value s up to a certain limit, $\alpha s \leq s_t$, which corresponds to the intuition that at most $(1-\alpha)\%$ of the light coming to this pixel can be reduced by a target shadow. A similar effect is expected for highlighted background, where the observed value is brighter than the expected value up to a certain limit.

In the our case, where the background is not static, there is no single expected value for each pixel. Let A be the sample values representing the background for a certain pixel, each represented as $x_i =< r_i, g_i, s_i >$ and, let $x_t =< r_t, g_t, s_t >$ be the observed value at frame t. Then, we can select a subset $B \subseteq A$ of sample values that are relevant to the observed lightness, s_t. By relevant we mean those values from the sample which if affected by shadows can produce the observed lightness of the pixel. That is, $B = \{x_i \mid x_i \in A \wedge \alpha \leq \frac{s_t}{s_i} \leq \beta\}$. Using this relevant sample subset we carry out our kernel calculation, as described in section 2, based on the 2-dimensional (r, g) color space. The parameters α and β are fixed over all the image. Figure 8 shows the detection results for an indoor scene using both the (R, G, B) color space and the (r, g) color space after using the lightness variable, s, to restrict the sample to relevant values only. We illustrate the algorithm on indoor sequence because the effect of shadows are more severe than in outdoor environments. The target in the figure wears black pants and the background is gray, so there is no color information. However we still detect the target very well and suppress the shadows.

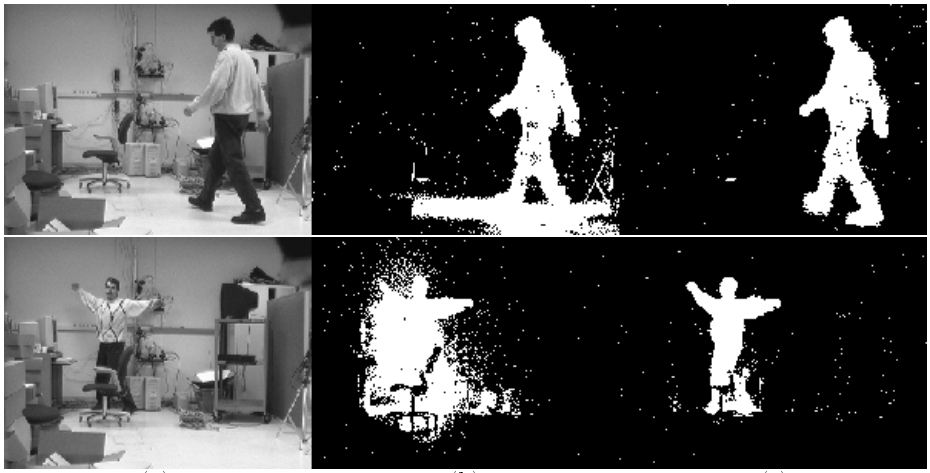


Fig. 8. (b) Detection using (R,G,B) color space (c) detection using chromaticity coordinates (r, g) and the lightness variable s

6 Comparisons and Experimental Results

6.1 Comparison

In this section we describe a set of experiments performed to compare the detection performance of the proposed background model as described in section 2 and a mixture of Gaussian model as described in [6,7]. We compare the ability

of the two models to detect with high sensitivity under the same false positive rates and also how detection rates are affected by the presence of a target in the scene.

For the non-parametric model, a sample of size 100 was used to represent the background; the update is performed using the detection results directly as the update decision, as described in section 2. For the Gaussian mixture model, the maximum number of distributions allowed at each pixel was 10[1]. Very few pixels reached that maximum at any point of time during the experiments. We used a sequence contains 1500 frames taken at a rate of 30 frame/second for evaluation. The sequence contains no moving targets. Figure 9 shows the first frame of the sequence.

Fig. 9. Outdoor scene used in evaluation experiments

The objective of the first experiment is to measure the sensitivity of the model to detect moving targets with low contrast against the background and how this sensitivity is affected by the target presence in the scene. To achieve this goal, a synthetic disk target of radius 10 pixels was moved against the background of the scene shown in figure 9. The intensity of the target is a contrast added to the background. That is, for each scene pixel with intensity x_t at time t that the target should occlude, the intensity of that pixel was changed to $x_t + \delta$. The experiment was repeated for different values of δ in the range from 0 to 40. The target was moved with a speed of 1 pixel/frame.

To set the parameters of the two models, we ran both models on the whole sequence with no target added and set the parameters of the two models to achieve an average of 2% false positive rate. To accomplish this for the non-parametric model, we adjust the threshold th; for the Gaussian mixture model we adjust two parameters T and α. This was done by fixing α to some value and finding the corresponding value of T that gives the desired false positive rates.

[1] this way the two models use almost the same amount of memory: for each distribution we need 3 floating point numbers a mean, a variance and a weight; for each sample in our method we need 1 byte

This resulted in several pairs of parameters (α, T) that give the the desired 2% rate. The best parameters were $\alpha = 10^{-6}, T = 98.9\%$. If α is set to be greater that 10^{-6}, then the model adapts faster and the false negative rate is increased, while if the α is less than this value, then the model adapts too slowly, resulting in more false positives and an inability to reach the desired 2% rate.

Using the adjusted parameters, both the models were used to detect the synthetic moving disk superimposed on the original sequence. Figure 10-a show the false negative rates obtained by the two models for various contrasts. It can be noticed that both models have similar false negative rates for very small contrast values; but the non-parametric model has a much smaller false negative rates as the contrast increases.

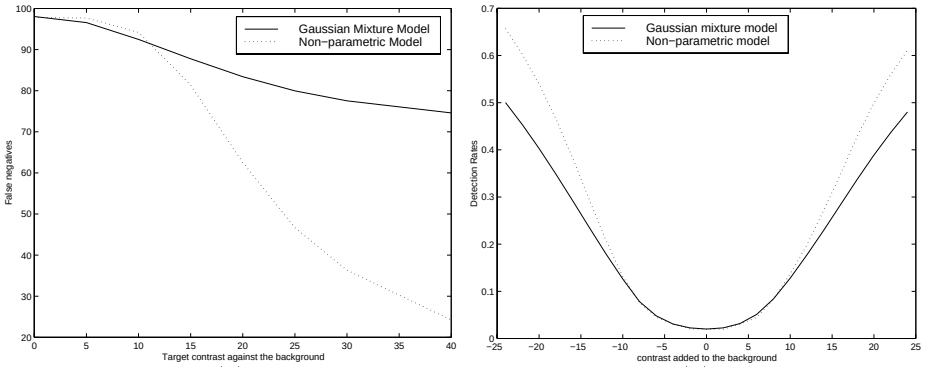


Fig. 10. (a) False Negatives with moving contrast target (b) Detection rates with global contrast added.

The objective of the second experiment is to measure the sensitivity of the detection without any effect of the target on the model. To achieve this a contrast value δ in the range -24 to +24 is added to every pixel in the image and the detection rates were calculated for each δ while the models were updated using the original sequence (without the added contrast.) The parameters of both the models were set as in the first experiment. For each δ value, we ran both the models on the whole sequence and the average detection rates were calculated, where the detection rate is defined as the percentage of the image pixels (after adding δ) that are detected as foreground. Notice that with $\delta = 0$ the detection rate corresponds to the adjusted 2% false positive rate. The detection rates are shown in figure 10-b where we notice better detection rates for the non-parametric model.

From these two experiments we notice that the non-parametric model is more sensitive in detecting targets with low contrast against the background; moreover the detection using the non-parametric model is less affected by the presence of targets in the scene.

6.2 Results

Video clips showing the detection results can be downloaded in either MPEG or AVI formats from ftp://www.umiacs.umd.edu/pub/elgammal/video/index.htm. Video clip 1 shows the detection results using 100 background samples. The video shows the pure detection result without any morphological operations or noise filtering. The video clip 2 shows the detection results for a color image sequence. Figure 11-top shows a frame from this sequence. Video clip 3 shows the detection results using both a short-term and a long-term model. The short-term model contains the most recent 50 background samples while the long-term contains 50 samples taken over a 1000 frame time window. Figure 11-bottom shows a frame from this sequence where the target is walking behind trees and is occluded by tree branches that are moving.

Fig. 11. Example of detection results

Video clip 4 shows the detection result for a sequence taken using an omni-directional camera[2]. A 100 sample short-term model is used to obtain these results on images of size 320x240. One pass of morphological closing was performed on the results. All the results shows the detection result without any use

[2] We would like to thank T.E. Boult, EECS Department, Lehigh University, for providing us with this video

of tracking information of the targets. Figure 12-top shows a frame from this sequence with multiple targets in the scene.Video clip 5 shows detection result for outdoor scene on a rainy day. The video shows three different clips for different rain conditions where the system adapted to each situation and could detect targets with the high sensitivity even under heavy rain. Figure 12-bottom shows a frame from this sequence with a car moving under heavy rain.

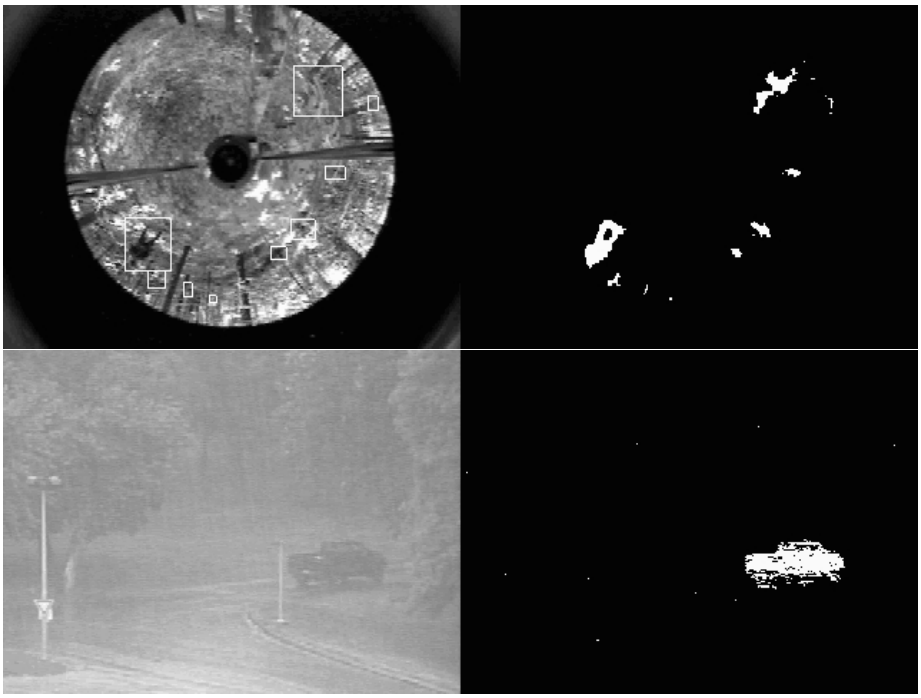


Fig. 12. Top:Detection result for an omni-directional camera. Bottom:Detection result for a rainy day.

7 Conclusion and Future Extensions

A robust, non-parametric background model and background subtraction mechanism that works with color imagery was introduced. The model can handle situations where the background of the scene is not completely static but contains small motions such as tree branch motion. The model is based on estimating the intensity density directly from sample history values. The main feature of the model is that it represents a very recent model of the scene and adapts to charges quickly. A second stage of the background subtraction was presented to suppress false detection that are due to small motions in the scene background based on

spatial properties. We also showed how the model can use color information to suppress shadows of the targets from being detected. A framework was presented to combine a short-term and a long-term model to achieve more robust detection results. A comparison between the proposed model and a Gaussian mixture model [6, 7] was also presented.

The implementation of the approach runs at 15-20 frame per second on a 400 MHz pentium processor for 320x240 gray scale images depending on the size of the background sample and the complexity of the detected foreground. Precalculated lookup tables for kernel function values are used to calculate the probability estimation of equation 5 in an efficient way. For most image pixels the evaluation of the summation in equation 5 stops after very few terms once the sum surpasses the threshold, which allows very fast probability estimation.

As for future extensions, we are trying to build more concise representation for the long term model of the scene by estimating the required sample size for each pixel in the scene depending on the variations at this pixel. So, using the same total amount of memory, we can achieve better results by assigning more memory to unstable points and less memory to stable points. Preliminary experiments shows that we can reach a compression of 80-90% and still achieve the same sensitivity in detection.

References

1. C. R. Wern, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of human body," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1997.
2. K.-P. Karmann and A. von Brandt, "Moving object recognition using and adaptive background memory," in *Time-Varying Image Processing and Moving Object Recognition*, Elsevier Science Publishers B.V., 1990.
3. K.-P. Karmann, A. V. Brandt, and R. Gerl, "Moving object segmentation based on adabtive reference images," in *Signal Processing V: Theories and Application*, Elsevier Science Publishers B.V., 1990.
4. D. Koller, J. Weber, T.Huang, J.Malik, G. Ogasawara, B.Rao, and S.Russell, "Towards robust automatic traffic scene analyis in real-time," in *ICPR*, 1994.
5. N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Uncertainty in Artificial Intelligence*, 1997.
6. W.E.L.Grimson, C.Stauffer, and R.Romano, "Using adaptive tracking to classify and monitor activities in a site," in *CVPR*, 1998.
7. W.E.L.Grimson and C.Stauffer, "Adaptive background mixture models for real-time tracking," in *CVPR*, 1999.
8. D. W. Scott, *Mulivariate Density Estimation*. Wiley-Interscience, 1992.
9. M. D. Levine, *Vision in Man and Machine*. McGraw-Hill Book Company, 1985.
10. E. L. Hall, *Computer Image Processing and Recognition*. Academic Press, 1979.