Background Maintenance Utilizing Common Distributions

by

Henry Hornblower Atkins III

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2003

Author .....
Department of Electrical Engineering and Computer Science
May 27, 2003

Certified by.....
W. Eric L. Grimson
Gordon Professor of Medical Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Background Maintenance Utilizing Common Distributions

by

Henry Hornblower Atkins III

Submitted to the Department of Electrical Engineering and Computer Science
on May 27, 2003, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Backgrounding is the process of maintaining a background model of a scene and using it to detect foreground objects within the scene. Backgrounding is a useful first step for many tracking and detection algorithms. A background maintenance algorithm that is similar to Stauffer and Grimson [5] except that it attempts to exploit relationships between the colors in a scene as feedback to the pixel process. The goal is that this common distributions approach will enable more accurate localization of foreground objects because it can better estimate the colors in a scene and their behaviors.

Thesis Supervisor: W. Eric L. Grimson
Title: Gordon Professor of Medical Engineering

Acknowledgments

Prof. Eric Grimson has been a pleasure to work with. Once, after a meeting with him, I ran into another student who asked how I was. I responded by saying, "I'm doing very well. I just finished meeting with Eric. I love meeting with Eric. He is always supportive of the work I have done and always offers useful and insightful suggestions. It makes my entire week look brighter." His response was: "I know exactly what you mean." Now, if I had just capitalized on all of the support that he gave me, I wouldn't be writing this the night before that absolute last possible day I can hand it in.

Chris Stauffer has in many ways the same effect that Eric has. I always feel excited and renewed about my work and everything else after talking with him. He has been invaluable in helping me implement my approaches. In particular, he is great for just chatting about ideas and life with.

Andy Sheaff is a great friend and lets me call him up at anytime to bounce ideas around. He always forces me to analyze all of the implications of any new approach. Besides, he doubles as a proofreader.

Kinh Tieu and Lily Lee (along with Chris) helped me realize the original topic for this thesis (even though they may not have noticed).

Finally, I have to thank all of my students who offered to write a chapter for me in exchange for higher project grades. Then of course there is my uncle Chester who offered to write my entire thesis claiming "If I write it for you, they will always remember you at MIT!"

Contents

List of Figures

Chapter 1

Introduction

Object tracking has many applications: automated systems for visual surveillance and security, tracking players during sporting events, or monitoring traffic flow rates and patterns. Utilizing vision for object tracking is particularly difficult. Objects may or may not be moving, they can change shape dramatically in a short time, and even if both motion and shape are held constant there may be multiple types of objects to track (cars, people, bikes, dogs, trucks). Furthermore, the scene may change. In outdoor scenes, the sun and clouds move, affecting light and shadowing. In indoor scenes, doors can be opened and closed, lights may be turned on and off, and light through windows may affect the scene just as it does in outdoor scenarios. In addition, in all situations, static objects may be moved into or out of a scene. For instance, a car may be driven into a parking lot, then left there. A chair may be placed in or repositioned within a room. Finally, there is camera noise and slight variation of otherwise static objects (e.g. waving trees, construction flashers). Such changes in the scene on top of the activities of the tracked objects require robust initialization for any tracking approach.

There are several approaches to initialization for visual tracking. One is to search the scene for the objects to track. However, such a 'detection by recognition' approach restricts the objects you can track to those for which you search. A more general approach to object localization is backgrounding.

Backgrounding is the process of differencing an image of a scene with no fore-

ground objects (a baseline or background image) from an image with objects present. The goal is to generate a mask image in which potential objects to track are highlighted. In the more general case, the baseline is not simply an image but a more thorough model of the background. Background maintenance is the process of creating and maintaining the background model. The entire process of backgrounding involves background maintenance as well as a method for determining how a new input deviates from the background model.

This process is complicated by several factors including: camera noise, lighting variations, moving background objects (e.g. fluttering objects), and static objects that are added, removed, or repositioned. This mask image highlights the locations of foreground objects. Therefore it may be input directly to a tracking engine or serve as the initialization for a specific object detector.

However, despite the difficulties, there are many benefits to backgrounding. It can be robust to camera noise. Backgrounding can be resilient to environmental changes as described previously. Finally backgrounding may be used to locate a moving object regardless of its appearance.

Backgrounding generally makes three assumptions about a scene. The first is that the objects to track are moving the majority of the time. The second is that over any sufficiently long window of time, a background pixel will be unobscured by foreground objects for the majority of the period. The final assumption is that there are detectable color differences between background and foreground pixels (i.e. the object can not be camouflaged). There are no other assumptions about the behavior, size, or shape of the objects to track.

1.1 Difficulties with backgrounding

There are several hurdles to overcome with background maintenance. Toyama et al. [7] laid out ten canonical problems:

> The first is the 'moved object' problem. If an inanimate object is repositioned within a scene should it be background or foreground? Because

we generally are not interested in tracking such an object, it should be background. However, because color and motion are the only cues to the background mechanism, such objects are generally considered foreground at least temporarily.

The second is the 'time of day' problem. Primarily in outdoor scenes but also in indoor situations, the movement of the sun and long-term variations in cloudcover lead to gradual yet pronounced changes in illumination and the locations of shadows or bright regions. Again, such changes should not affect the notion of what is background but they constitute color and motion cues and thus can confuse many background approaches.

The third is the 'light switch' problem. This refers to sudden global scene changes. For instance, turning on lights in a room. Such variations affect the entire scene and all of the objects at once.

The fourth problem is 'waving trees.' Frequently background objects move. Waving trees or flapping flags are the standard examples however construction flashers cause the effect. Such objects exhibit cues that may lead a background subtraction approach to label them as foreground however that is generally not the desired behavior. The one cue that generally sets them apart from interesting objects is the periodicity and range of motion. Tree branches waving in the breeze have a high frequency and low range relative to a person pacing in front of a building.

The fifth problem is that of 'camouflage.' A person dressed in white walking in front of a white building should be foreground. However, that event probably does not trigger the color and motion cues necessary for a background approach to correctly label it.

The sixth problem is how to initialize a background model with the presence of foreground objects. 'Bootstrapping' is a difficulty in many surveilled locations. For instance, how do you make all the cars stay off the road so you can take a background picture?

The problem of 'foreground aperture' arises when a uniformly colored

foreground object moves. The motion cues only appear at the edges of the object. Thus background subtraction only detects the edges of the objects as foreground.

The eighth is the 'sleeping person' problem. This is akin to the 'moved object' problem. When an object of interest, such as a person, enters a scene and remains still, it is desirable to continue to detect that person as foreground.

The ninth problem is 'waking person' which mirrors the sleeping person problem. When a foreground object such as a parked car has been still in a scene and begins to move, it should be considered foreground. However, the region of background that it uncovers is frequently also considered foreground.

The final problem is 'shadowing.' With a point light source, objects cast shadows. Those shadows should not be part of the object however they are generally regarded as such by the backgrounding algorithm.

Most of these problems arise because of a 'poverty of the stimulus.' The only input to the backgrounding process is the stream of images and as a result, the previous background model. Any other input requires a priori knowledge of the objects of interest or the scene.

1.2 Background of Backgrounding

Backgrounding is a popular first step for visual tracking. The simplest approaches represent each pixel in the background model by its average value over time. More complex approaches utilize filters or other predictive methods to estimate the background model for each frame.

Wren et al. [8] represent each background pixel process with a single Gaussian. Their system is specifically designed for a static background and makes no attempt to solve the bootstrapping problem. They use an initialization period in which there are no foreground objects in the scene.

Stauffer and Grimson [5] utilize a mixture of Gaussians to represent every pixel process. Each Gaussian is updated with a weight that is an estimate of the degree to which that Gaussian represents the recent history of the pixel. By allowing multiple distributions per pixel, they can accurately account for items such as construction flashers as part of the background model.

Friedman and Russel [1] present a similar approach using Gaussian distributions. However, instead of using a general mixture as Stauffer and Grimson [5] they attempt to classify each pixel into one of either vehicle, road, or shadow distributions. There approach was designed for the particular application of tracking vehicles on a segment of road.

Harville et al. [3] add the additional information of depth to the standard mixture of Gaussians approach. Unfortunately, their approach requires stereo cameras.

Toyama et al. [7] utilize a layered approach to background maintenance. At the pixel level, they represent each pixel process with a Weiner filter. This allows them to predict the pixel value in the next frame. Significant deviation from this prediction suggests the pixel is foreground. Their region level framework tries to segment out moving objects by detecting the motion at the edges of the object. Finally, their frame level processing decides when there has been a global scene change that requires a completely new background model.

Gutchess et al. [2] make use of optical flow information to determine which values are background. They do so by recognizing events where an object moves over a pixel value then later moves off of a pixel value. These covering and uncovering events allow them to determine which values from a pixel process were foreground and which were background. This approach is specifically designed to allow initialization of a background model while foreground objects are present in the scene ('bootstrapping').

Rittscher et al. [4] and Stenger et al. [6] use Hidden Markov Models to represent each pixel process. However, inherent in such a representation is a direct time dependence and they also assume independence between pixels. Stenger et al. [6] go one step further to use topology free HMMs. This allows the state to split (akin to the multimodality of backgrounds in Stauffer and Grimson [5]). This is convenient for

the global scene changes. In their testing, Stenger et al. [6] generate a background model for a train station that accounts for both the cases where the train is there and where it is not.

Previous work in backgrounding centers around either trying to more accurately represent the state of the background or trying to extract a little more information from a sparse world. Friedman and Russel [1], Harville et al. [3], Stauffer and Grimson [5], Stenger et al. [6], Wren et al. [8] attempt to accurately represent as much information as possible for each pixel. Gutchess et al. [2], Toyama et al. [7] attempt to make higher level assumptions about events that affect the pixel processes.

1.3 Goals of my approach

I propose an extension to Stauffer and Grimson [5] by sharing information across pixels thus providing some feedback and more support for estimating every pixel process. My hope is that this will enable faster, more accurate updates to the background model because the individual effect of each pixel can be smaller. However, with multiple pixels, the effect of a global lighting change will update the distribution faster. In addition, each distribution can maintain its own state about whether it is background or foreground. Thus, in the case of the 'waking person' problem the newly revealed background will have a high prior to be labeled as background because the pixels will fall under a distribution that is background at every other pixel. In the case of the 'sleeping person' problem, the feedback loop allows pixels on the object, as they decide to become background, to encourage other pixels on the object to do the same.

I propose an approach that will aggregate the information across multiple pixels to solve some of the problems laid out by Toyama et al. [7]. There are two implicit assumptions in such an approach. First is the assumption that the majority of the scene is background. Such is generally the case in far field tracking. Second is the assumption that the background occurs in reasonably contiguous uniform color regions.

Chapter 2

Approach

My approach is a direct extension of the work of Stauffer and Grimson [5]. Stauffer and Grimson [5] represented each pixel process with a mixture of Gaussians. At every time step, the model for each pixel is updated with the latest observation. In the common distribution framework (described in section 2.2), I propose some limited sharing of the observation information across pixels.

2.1 Mixture of Gaussians

Stauffer and Grimson [5] represent each pixel process with a mixture of Gaussians. In particular, the probability of the current pixel value is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2.1}$$

For a pixel represented by K distributions and the i^{th} distribution at time t: $\omega_{i,t}$ is the weight estimate in Equation 2.4, $\mu_{i,t}$ is the mean, and $\Sigma_{i,t}$ is covariance matrix. Finally, η is the Gaussian probability density function

$$\eta(X_t, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(X_t-\mu_t)^T\Sigma^{-1}(X_t-\mu_t)} \tag{2.2}$$

For computational convenience, the covariance matrix is assumed to be of the form:

$$\Sigma_{k,t} = \sigma_k^2 \mathbf{I} \tag{2.3}$$

This assumes independence between the color channels.

With every new pixel value X_t, the parameters are updated as follows: The new pixel is checked against the K Gaussian distributions. It is considered to match if X_t is within 2.5 standard deviations of the mean of a distribution. If the pixel does not match any of its distributions, the distribution with the lowest weight $\omega_{i,t}$ is replaced by a new distribution with mean X_t, a high variance, and a low weight. By construction, the pixel will match this new distribution.

After matching, the prior weights $\omega_{k,t}$ of the K distributions are updated as follows

$$w_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{2.4}$$

where $M_{k,t}$ is 1 for the distribution that matched and 0 elsewhere. α is a constant and $1/\alpha$ is the time constant for the update of the parameters of the distributions. Finally, the weights are re-normalized so that they sum to 1.

For all unmatched distributions, the parameters μ and σ remain unchanged, however the parameters of the matching distribution update as

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{2.5}$$

$$\sigma_t^2 = (1 - \rho) * \sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{2.6}$$

and

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \tag{2.7}$$

Background pixels should have relatively static values. That means that they have low variance and match with a Gaussian of high weight. To decide which pixels should be labeled as background, Stauffer and Grimson [5] begin by sorting the Gaussians in descending order by ω/σ. Thus pixels that have been represented primarily by a single

distribution and have a low variance are more likely to be considered background. To implement this, the first B distributions for each pixel are chosen as the background model where

$$B = argmin_b(\sum_{k=1}^{b} \omega_k > T) \tag{2.8}$$

and T is a threshold representing the minimum portion of the data that should be accounted for by the background. A small T restricts the background to a unimodal distribution whereas a larger T accepts more distributions thus allowing a multimodal background model.

2.2 The Common Distributions Approach

I propose some extensions to the work of Stauffer and Grimson [5]. The primary goal is to capitalize on the repetition of information across pixels by allowing a single Gaussian to represent multiple pixels. Each pixel will continue to have an independent set of K Gaussians that account for its history with weights $\omega_{k,t}$. However, instead of matching against only those K Gaussians, each pixel is matched against all Gaussians. Therefore the pixels in a region of uniform color will potentially all map to a single Gaussian.

There is a set of Gaussians $\mathcal{G} = \{G_1, \cdots G_N\}$ where each Gaussian G_i at time t is represented by a mean $\mu_{i,t}$ and covariance $\Sigma_{i,t}$.

$$G_{i,t} = \{\mu_{i,t}, \Sigma_{i,t}, F_{i,t}\} \tag{2.9}$$

F is an estimate of the foreground probability of the Gaussian (this will be discussed later). $\Sigma_{i,t}$ is assumed to be diagonal but unlike Equation 2.3 the variances of each channel are not assumed to be equal.

There is also a set of pixels $\mathcal{P} = \{P_1, \cdots P_M\}$ where each pixel P_i at time t consists of its position, most recent color observation $X_{i,t} = \{r_{i,t}, g_{i,t}, b_{i,t}\}$, and a set

of K references $\{g_1 \cdots g_K\}$ to Gaussians in $\mathcal{G}$.

$$P_{i,t} = \{\{x_i, y_i\}, X_{i,t}, g_{1,i,t} \cdots g_{K,i,t}\} \tag{2.10}$$

In each pixel, g_k is a reference to a Gaussian and has the form

$$g_{k,i,t} = \{G_j, \omega_{i,j,t}\} \tag{2.11}$$

where $\omega_{j,t}$ is the weight associating pixel P_i to Gaussian G_j as in Equation 2.4.

At every time step, each pixel in $\mathcal{P}$ is compared to every Gaussian in $\mathcal{G}$. A pixel P_i is considered to match a Gaussian G_j if G_j minimizes q defined as:

$$q = \left| \frac{|X_{i,t} - \mu_{j,t}|}{\Sigma_{j,t}} \right| \tag{2.12}$$

Thus q is the normalized squared distance from the pixel to the mean of the distribution. This match occurs subject to the constraint that $|X_{i,t} - \mu_{j,t}| < 2.5 \cdot \sigma_{j,t}$ as with Stauffer and Grimson [5]. If the pixel does not match with any Gaussian in $\mathcal{G}$ then a new Gaussian is added with an initially high variance and low weight.

For all unmatched Gaussians, the parameters μ and σ again remain unchanged. However, for all matched Gaussians the updates occur per Gaussian as

$$\mu_t = (1 - \phi\mu_{t-1} + \phi \mathrm{E}[X_{t,match}] \tag{2.13}$$

$$\Sigma_t^2 = (1 - \phi)\Sigma_{t-1}^2 + \phi(\mathrm{E}[X_{t,match} - \mu_t])^T(\mathrm{E}[X_{t,match} - \mu_t]) \tag{2.14}$$

where

$$\phi = min\left(\frac{P_{t,match}}{C}, .9\right) \tag{2.15}$$

$P_{t,match}$ are all the pixels that matched at time t with the Gaussian being updated, and $X_{t,match}$ are their current values, and C is some suitable constant that is primarily dependent upon the size of the input image. The purpose of ϕ is to modulate the update of the Gaussians. Gaussians with one pixel of support will update slowly but

Gaussians with more support will update faster up to a point. The assumption is that with more samples the estimate of the color is more likely to be accurate.

For purposes of creating a background model, each pixel still maintains its own set of Gaussians to which it matched with weights updated as in Equation 2.4. The difference now is that instead of using this information to generate a binary decision about background versus foreground, I generate a confidence $f_{i,t}$ which is an estimate of how likely pixel P_i is to be background at time t. For this purpose, if pixel P_i matched with Gaussian G_j

$$f_{i,t} = (1-\gamma)\frac{\omega_{i,j,t}}{max(\omega_{i,t})} + \gamma F_{j,t-1} \tag{2.16}$$

Finally, because of the one-to-many mapping of Gaussians to pixels, we can determine a probability that a Gaussian represents a background color $F_{j,t}$. For any Gaussian G_j at time t,

$$F_{j,t} = (1-\beta) * F_{j,t-1} + \beta * \mathrm{E}[f_{match,t} > T_{background}] \tag{2.17}$$

Thus at time t, every pixel has an initial estimate of whether it is background and every Gaussian has an estimate from time $t-1$ of whether it is background. By combining these two estimates we can hopefully achieve a more accurate determination of whether a pixel is background or foreground. Because these are two estimates of the probability of an event, they can be averaged with some appropriate weighting to estimate the probability that a particular pixel is background. That weighting is determined by γ. $T_background$ is a threshold on the probability above which a pixel is considered background.

2.3 Proposed Benefits of the Common Distributions Approach

I hope that there are several advantages to this approach of sharing color models across pixels. In particular, I hope to better estimate the color model of an object by matching multiple pixels to a single Gaussian. In addition, by allowing the Gaussians to impart feedback upon the pixels, I hope to better estimate the probability that a pixel is background.

The common distribution approach that I have described creates a global color model of a scene. This model consists of the set of Gaussians $\mathcal{G}$. Each Gaussian has a mean value, variance, and probability that it is background at any particular time. As a result, if a foreground object reenters the scene, the system will already have an accurate color model for it.

Again because of the shared color model, pixels in a uniform color region will generally match with the same Gaussian. Therefore, there is a larger sample set from which to estimate the parameters of the distributions. A larger sample suggests that I can place more confidence in the new estimates and thus update the color model at a higher rate.

The 'waking person' problem is common to backgrounding. When a previously stationary object moves, it is beyond the scope of the backgrounding problem to determine that the uncovered region should be background. One approach would be to apply higher level assumptions and treat it as an 'uncovering' event from Gutchess et al. [2]. In most backgrounding however, because each pixel is independent, the uncovered region dissolves piecemeal into the background over time.

In my approach, there are potentially two advantageous factors in such a situation. Assuming that the uncovered region is uniform, it should dissolve into the background all at once. As pixels decide that they are background, the bias of the color will shift towards an estimate that it is background and other pixels will be faster to follow suit. The unfortunate side-effect, is that before the color transitions to background it will be biased to the foreground and thus will exert pressure on the matching pixels to

stay foreground. This problem is mitigated however when the revealed pixels match a distribution that is already background. For instance, if the revealed background pixels where part of a parking lot that was already considered background.

Finally, for the same reason that revealed background of a novel color may have a tendency to remain as foreground, I hope that my approach will more resilient to background objects that are moved and foreground objects that are stopped (ie. the 'moved object' and 'sleeping person' problems). If a chair is repositioned in a room, under most approaches it would be foreground and be re-incorporated into the background over time. However, in the case of a single chair, it's color model would bias it towards becoming background sooner. More importantly, in a room full of similar chairs, if one is repositioned all of the other chairs that match its color model will tend to bias it to be background. This is analogous to the situation of revealed pavement described above. In addition, if a foreground object pauses, it's color model will continue to bias it towards being in the foreground. In a perhaps less likely scenario, if there was a group of foreground objects that shared a color model (e.g. a soccer team with identical jerseys) the situation would be identical to that of the moved chair.

2.4 Potential Pitfalls of the Common Distributions Approach

My primary concern with this approach is that it is non-deterministic. At it's heart, it is a clustering algorithm and as such its results are dependent upon how it is initialized. For instance, on the first iteration a particular pixel P_i may match Gaussian G_j. However, a later pixel P_k may not match any of $\mathcal{G}$ and thus a new Gaussian G_l will be created. The problem arises when G_l is a better match for P_i by Equation 2.12 than the original match G_j. Fortunately, this is not a serious problem in practice. The updating of the weights Equation 2.4 makes the single match mistake insignificant. However, this can occur in the first iteration of matching. At that point, because the

weights are normalized, the incorrect match G_j will have a weight of 1.0 with pixel P_i and it will take many frames for that to decay away. The solution I have implemented is to perform an initial pass to generate $\mathcal{G}$ before beginning the process of matching.

Another potential problem is that the histories for particular pixels may be invalidated over time. This is because a single Gaussian can represent multiple different pixels at different times. For instance, at some time $t - \delta$, pixel P_i matched with Gaussian G_j where δ is some arbitrary delay such that G_j is still in the history for P_i. At time t these two objects may no longer match. In such a situation, in the interim δ, G_j may have matched with other pixels. Through updates in those iterations, G_j may no longer accurately represent the state of P_i at time $t - \delta$. The models of pixels over time may no longer be valid. Thus if pixel P_i reverts to its value at $t - \delta$ it will no longer match G_j and be erroneously labeled as foreground.

Finally, the entire system may be unstable. At every time step, the Gaussians update their probability of being background using the background probabilities of the pixels that matched. However, on the following iteration, the pixels make decisions again about whether they are background. This decision is determined in part from the background probability of the Gaussian from the previous time step.

2.5 Summary of the Common Distributions Approach

I propose a background maintenance algorithm that is similar to Stauffer and Grimson [5] except that I am trying to exploit dependencies between pixels. Whereas they utilized an independent mixture of Gaussians to represent every pixel process, I propose maintaining a separate mixture per pixel but that the Gaussians be shared across all pixels. On each iteration, every pixel will be compared against every Gaussian and will be matched with the one to which it is closest. The quality of a match is measured by the normalized squared distance from the mean of the distribution to the pixel observation. After matching, the weights associating each pixel to the

recent history of Gaussians to which it matched are updated as in Stauffer and Grimson [5]. The probability that a single pixel is background is determined by Equation 2.16 which allows a pixel to have a multimodal background model. The probability that a Gaussian is background is determined by an online EM update (Equation 2.17) utilizing the expectation that the pixels it previously matched with were background. Then each pixel is labeled as background or foreground by thresholding the weighted average of the pixel's estimate and the matching Gaussian's estimate. Finally, the Gaussian mean and variance parameters are updated in batch from the pixels with which it matched.

Chapter 3

Results

3.1 Testing

To test my approach, I compared it to an implementation of Stauffer and Grimson [5]. However, because the intent of the comparison was to analyze the differences between the two approaches (namely sharing the distributions versus not sharing them) it is not intended to be a complete and tuned implementation of Stauffer and Grimson [5]. In particular, ρ (Equation 2.7) is implemented such that $\rho = \alpha$ instead of as a function of η, the Gaussian pdf (Equation 2.2). The results of both algorithms have been postprocessed to eliminate all 4-connected-components smaller than 4 pixels as suggested by Stauffer and Grimson [5].

Figure 3-1 shows some images from the test data. The test dataset was from the PETS2001 workshop. It is a single camera view of a parking lot with some pedestrian and automobile traffic. I ran both the mixture of Gaussians Stauffer and Grimson [5] and the common distributions approach on the dataset with a variety of parameters. As an input sequence, I used every tenth image of the original video sequence. The output sequences are binary images where white represents pixels that were labeled as foreground.

There are four output sequences representing the best of several runs: two from each approach with different parameters. Figure 3-2 illustrates the results of the common distributions approach with a $T_{background} = .75, \gamma = .2, \alpha = .05$ and the

initial variance of a new Gaussian was .005. Figure 3-3 is another run of the common distributions approach but instead the default variance of a new Gaussian was .001. The difference in noise is immediately clear. The lower default variance means that each Gaussian represents more pixels and a larger range of colors.

Figure 3-4 shows the output from the mixture of Gaussians approach with $T = .4, \alpha = .05, K = 3$ and new Gaussians have a default variance of .05. Figure 3-5 is a second iteration of mixture of Gaussians but with a default initial variance of .001. Again, the difference in the noise is obvious.

3.2 Results

The first image in the sequence is from a period of time with no activity. It simply illustrates the effect of noise on each of the backgrounding algorithms. Clearly, the image is noisier in the darker regions with less uniform color. Both mixture of Gaussians implementations handle this situation extremely well. For common distributions, Figure 3-2 has less noise than Figure 3-3 because it uses a higher default variance. As a result, it is representing the scene with nearly half as many Gaussians. Each of which has a larger variance and is thus less sensitive to slight variations in the individual pixel values.

The second image has a car entering from the left and the car parked nearest the middle of the scene is backing out of a parking space. All of the illustrated approaches detect the motion of the reversing car. However, the common distribution approach detects more of the object because regions of uniform color are pulled to the foreground as a group. Additionally, the mixture of Gaussians approaches appear to be a little more sensitive to the noise of the pixels surrounding the car approaching from the left.

In the third image, the reversing car has stopped while the other vehicle continues to approach from the left. Again, all of the algorithms have no trouble with this simple motion. The mixture of Gaussians approach again appears to be slightly more sensitive to the noise. Another point of interest is that all approaches have lost track

of the reversing car. This illustrates the 'moved object' problem. The car was still and thus part of the background, then it moved slightly and became still again. All approaches have successfully reinserted it into the background.

By the fourth frame the car from the left has stopped before backing into a space and the reversing car in the middle has continued its move from its parked location. Figure 3-4 shows that this implementation of mixture of Gaussians has lost track of the car from the left just as it pauses to prepare to reverse into an empty space. This is an instance of the 'sleeping person' problem. The car has paused long enough such that the mixture of Gaussians approach has incorporated it into the background whereas the feedback built into the common distributions approach has allowed it to hold onto the car for longer.

In the fifth frame, there is an illustration of the 'waking person' problem. The space obscured by the parked car in the center is revealed as the car backs out. The mixture of Gaussian approaches seem to have lost track of the car from the left that is now parking. Additionally, in Figure 3-4 appears to have labeled a swath of the parking space as foreground. However, the common distribution approach seems to have labeled more of the parking space as foreground. This is because the parking space is painted and thus does not fit the color model of the rest of the pavement. In this frame a pedestrian has entered from the right and is detected by all of the approaches.

In the sixth, seventh, and eighth images, the two pedestrians continue across the scene to the left. As they appear smaller they become more difficult to detect. All approaches show difficulty detecting such small objects however the common distributions approaches seem to do a slightly better job of selecting pixels on the pedestrians.

Finally, in the remaining two images, a bicyclist is moving across the scene to the left. One difficulty here is that the wheels of the bike are very thin and the spoke regions appear nearly transparent. In this case, both of the common distributions approaches appear to do a better job of segmenting the cyclist including the wheels of the bike.

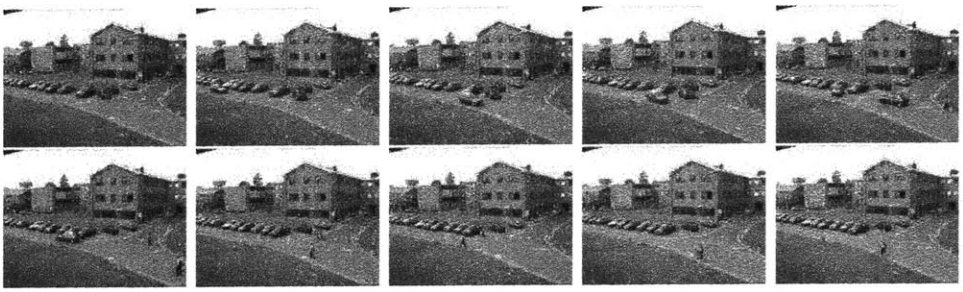

Figure 3-1: A series of images from the input data.

In all of the approaches, an increase in the detection probability leads to an increase in false detections as is to be expected. The common distribution approach appears to more uniformly detect an object as foreground without missing individual pixels and leaving holes in the mask. This is a result of the dependencies between pixels of similar color. Perhaps most notably, we can see some of the problems described by Toyama et al. [7]. The problem of backgrounding is potentially under constrained. In the third image, the reversing car paused and was reimplemented into the background as if it was a repositioned background object. However, with additional information, we know that the car should have been foreground. Unfortunately there is not necessarily information available to the background algorithm to make such a decision.

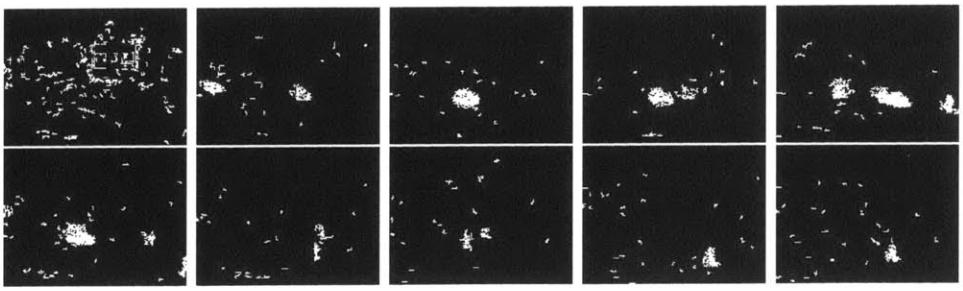

Figure 3-2: The results of the common distributions approach. Default variance = .005

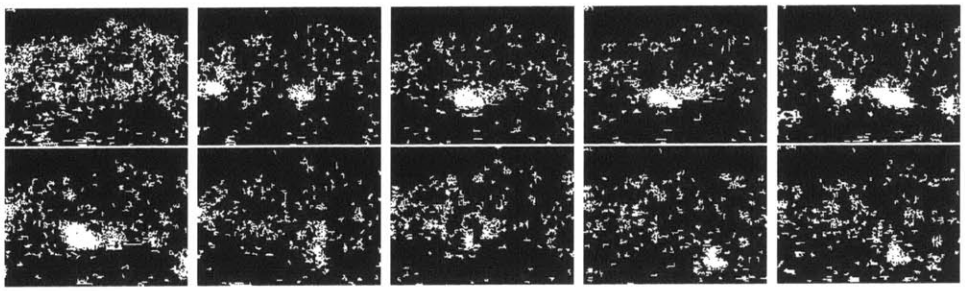

Figure 3-3: The results of a the common distributions approach. Default variance = .001

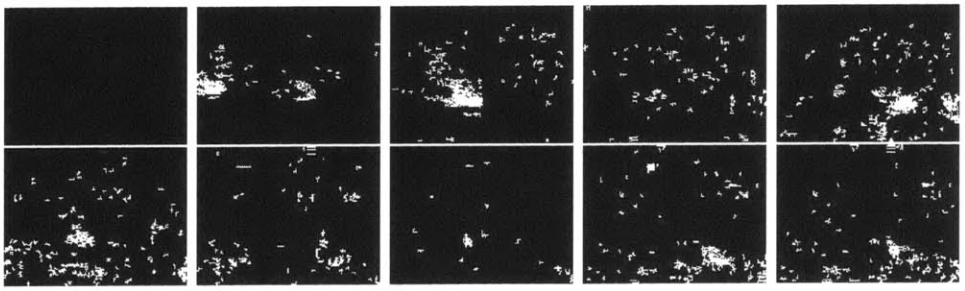

Figure 3-4: The results of the mixture of Gaussians approach Stauffer and Grimson [5]. Default variance = .05

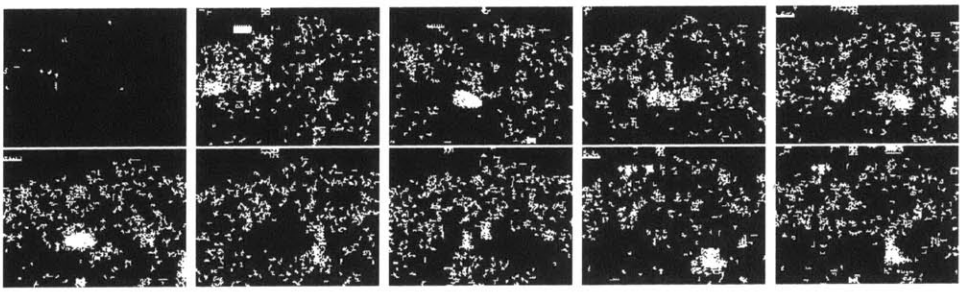

Figure 3-5: The results of the mixture of Gaussians approach Stauffer and Grimson [5]. Default variance = .001

Chapter 4

Conclusions

The common distributions approach for background maintenance illustrates some of the promise of backgrounding as well as some of the inherent difficulties. It can accurately label regions of activity however it can also mislabel regions of noise. Perhaps most importantly, it can misclassify regions because it makes assumptions about the state of the world which are not necessarily true.

The common distributions approach labels foreground objects more uniformly (without 'holes') because it assumes a dependency amongst pixels of similar colors and that objects tend to be relatively uniform in color. This feature allows it to more accurately label novel objects because of assumed relationships via color information (ie. it is a new color not yet in the scene). However, this very much relies on the assumption that the foreground objects are not the same color as the background objects. Otherwise, the system works against itself.

Unfortunately, some of the features of the common distributions approach that are designed to allow it to solve more problems in backgrounding also potentially cause some more. For instance the reintegration of the paused car into the background as though it was a moved object. Additionally, the revealed background that was kept as part of the foreground model for longer because it was a relatively novel color. It seems to support the claims of Toyama et al. [7] that backgrounding might not be the appropriate level to solve classification problems because of the poverty of the stimulus. Perhaps backgrounding should be at most a coarse filter to a recognition

or tracking engine.

An unfortunate drawback to the common distributions approach is its speed or lack thereof. Whereas [5] has to test for K matches per pixel, the common distributions approach has to test for $|\mathcal{G}|$ matches per pixel. In the two examples illustrated previously, $|\mathcal{G}|$ (the number of Gaussians) was about $130-250$. This does not compare favorably with [5] in which K varies from $3-5$.

In addition, I initially had concerns that the feedback in the system would make it unstable. This is not a problem provided $\gamma + T_{\mathit{background}} < 1$. Otherwise the entire scene is labeled as foreground.

Any benefits of the common distributions approach are because it assumes some dependence between pixels of similar colors and uses them to generate more accurate color models. However, these benefits may also cause problems because they make more assumptions about the world.

Despite this, I feel that the common distributions approach does represent an improvement (however small) over the mixture of Gaussians approach upon which it is based. It more completely highlights objects and is more robust to some camouflage problems.

Although backgrounding is potentially poorly defined, it still serves as a useful initialization step for many tracking algorithms. Backgrounding relies on some assumptions about the world that generally hold. Such as pixels maintaining a relatively consistent color unless a foreground object is present and that foreground objects are generally in motion. Over the range of situations in which we generally track, such assumptions hold reasonably well and thus background can be a useful tool.

Bibliography

[1] N. Friedman and S. Russel. Image segmentation in video sequences: A probabilitic approach. In *Proc. of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 1997.

[2] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A. K. Jain. A background model initialization for video surveillance. In *Int. Conf. Computer Vision*, 2001.

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

[4] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. In *European Conf. on Computer Vision*, 2000.

[5] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. Computer Vision and Pattern Recognition*, pages 246–252, 1999.

[6] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. M. Buhmann. Topology free hidden markov models: Application to background modeling. In *Int. Conf. Computer Vision*, 2001.

[7] K. Toyama, J. Krumm, B. Brummit, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proc. International Conference on Computer Vision*, Corfu, Greece, 1999.

[8] C. Wren, A. Azerbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.