PROBABILISTIC METHODS FOR ADAPTIVE BACKGROUND SUBTRACTION

J. MIKE MCHUGH

Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

BOSTON UNIVERSITY

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Thesis

PROBABILISTIC METHODS FOR ADAPTIVE BACKGROUND SUBTRACTION

by

J. MIKE MCHUGH

B.S., Worcester Polytechnic Institute, 2006

Submitted in partial fulfillment of the

requirements for the degree of

Master of Science

2008

Approved by

First Reader

Janusz Konrad, Ph.D.
Professor of Electrical and Computer Engineering

Second Reader

David A. Castañon Ph.D.
Professor of Electrical and Computer Engineering

Third Reader

Venkatesh Saligrama, Ph.D.
Professor of Electrical and Computer Engineering

Acknowledgments

First and foremost, I would like thank my advisor, Professor Janusz Konrad, for all of his support and guidance on this research and throughout my graduate studies. He has done an outstanding job of teaching me, and moreover, leading me in the right directions so that I may learn to better teach myself. I am also grateful for all of the times that, when I've over analyzed and overburdened myself, he has kept me honest and realistic.

In addition, many thanks go out to collaborators on this research, Professor Venkatesh Saligarama at Boston University and Professor Pierre-Marc Jodoin at University of Sherbrooke. Learning from the wisdom of these men has been paramount to my understanding of this work.

Thanks to the students in the ISS (Information Systems and Sciences) group at BU. Deserving of special notice are Lung-Chang (Jack) Hsieh, Serdar Ince, Erhan Ermis, Hossein Entekhabi, and Liz Begin for their collaboration with me on this research.

Finally, thanks to my committee members Janusz Konrad, David Castañon, and Venkatesh Saligrama for reading this thesis and providing your comments.

PROBABILISTIC METHODS FOR ADAPTIVE BACKGROUND SUBTRACTION

J. MIKE MCHUGH

ABSTRACT

Visual surveillance applications such as object identification, movement tracking, and activity monitoring require reliable moving object detection as an initial processing step. The process that segments moving objects from the stationary scene is termed background subtraction. In the past, the most effective background subtraction methods have been those that employ probabilistic modeling of the background. In this thesis, we present three adaptive detection modalities that apply generally to any pixel-based probabilistic background model. Formulated within the classical binary hypothesis framework, a method for explicit foreground modeling is proposed, which refines the decision process wherever moving objects are present and reduces the miss rate. Also, by modeling the detection mask as a Markov random field, we present a method that adapts spatially to impose continuity on the detection result. In addition, we apply a multiple comparisons procedure known as false discovery rate control. The result is a temporally variable thresholding strategy that adapts to object sparsity. The new methods offer a clear qualitative improvement in real scenarios as well as a measurable performance gain over non-adaptive techniques when tested on synthetic sequences.

Contents

List of Tables

List of Figures

List of Abbreviations

CDF	....	Cumulative density function
DFT	....	Discrete Fourier transform
FDR	....	False discovery rate
FFT	....	Fast Fourier transform
FNR	....	False negative rate
FPR	....	False positive rate
ICM	....	Iterated conditional modes
IDFT	....	Inverse discrete Fourier transform
LIS	....	Local-in-space
LIT	....	Local-in-time
LRT	....	Likelihood ratio test
MCP	....	Multiple comparisons procedures
MoG	....	Mixture of Gaussians
MRF	....	Markov random field
PC	....	Phase correlation
PDF	....	Probability density function
PTZ	....	Pan-tilt-zoom
ROC	....	Receiver operating characteristics
TER	....	Total error rate

Nomenclature

$\mathbb{R}^2$	....	the Real plane
Λ	....	Image pixel domain sampling lattice
Γ	....	Correlation lag domain sampling lattice
$\boldsymbol{m}, \boldsymbol{n} = [n_1\ n_2]^T$	....	Discrete space index; image pixel domain
$\boldsymbol{d} = [d_1\ d_2]^T$	....	Discrete space index; correlation lag domain
$\boldsymbol{u} = [u_1\ u_2]^T$	....	Discrete frequency index; discrete Fourier domain
k, l	....	Discrete time index
$I[\boldsymbol{n}]$	....	Intensity image
$C[\boldsymbol{d}]$	....	Correlation surface
$\Psi[\boldsymbol{d}]$	....	Normalized correlation surface
$\mathrm{DFT}\{J[\boldsymbol{n}]\} = \widehat{J}[\boldsymbol{u}]$	....	2D discrete Fourier transform of J
$\mathrm{IDFT}\left\{\widehat{J}[\boldsymbol{u}]\right\} = J[\boldsymbol{n}]$	....	2D inverse discrete Fourier transform of $\widehat{J}$
$E[\boldsymbol{n}]$ (and $e[\boldsymbol{n}]$)	....	Random label field (and realization); a.k.a. detection mask
$\mathcal{B}$, and $\mathcal{F}$	....	Background label and foreground label
$\pi_{\mathcal{B}}$, and $\pi_{\mathcal{F}}$	....	Background prior probability and foreground prior probability
η	....	Cost term
θ	....	Threshold
$\Pr\{X\}$	....	Probability of random event X
$\mathbb{E}\{X\}$	....	Expectation of random quantity X
$\mathcal{P}$ (and p)	....	Random significance score (and realization)

Chapter 1

Introduction

Background subtraction is a general term for a process which aims to segment moving foreground objects from a relatively stationary background. As it has been noted in (Migdal and Grimson, 2005) and (Radke, 2005), we point out that that there is an important distinction between the *background modeling* and *background detection* stages, which comprise the whole subtraction process. As illustrated in Figure 1.1, these two stages are often interrelated and sometimes overlapping. The modeling stage creates and maintains a model of the background scene. The detection process is responsible for segmenting the current image into moving (foreground) and stationary (background) regions based on the current background model. The resulting detection masks may then be fed back into the modeling process in order to avoid corruption of the background model by foreground objects.

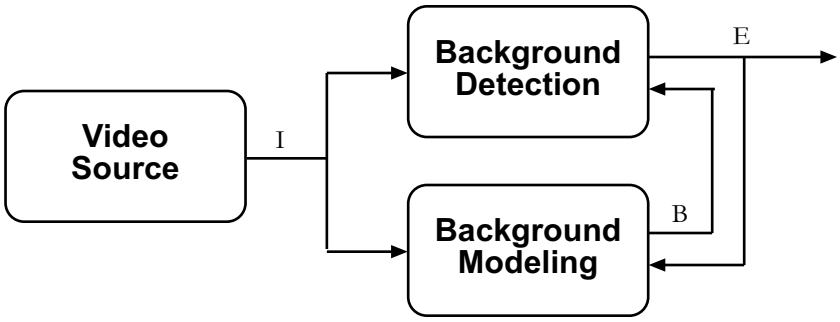

Figure 1.1: A video source produces images I. The modeling stage uses previous video frames and detection results to maintain background model B. The detection stage compares the current frame to the current model to produce a detection mask E.

It's worth noting that the task of background subtraction is different from other types of image change detection that compare two images taken at different times. Examples

of the latter may include object based video coding, remote sensing, or biomedical image applications. In our case, the scenario is visual surveillance with many images arriving at a high rate (around 25 frames per second typically). Generally, the setup involves a stationary surveillance camera monitoring some scene, which is traversed by objects such as cars on a road or pedestrians on a sidewalk. For any type of image change detection task, one aims to detect 'interesting' changes and to ignore 'uninteresting' ones (Radke, 2005). For background subtraction, the types of changes we wish to detect are those which correspond to moving objects - e.g., a car occupies a region that is normally the road. The types of 'uninteresting' changes, which we wish to be robust to, include global motion due to camera shake; gradual illumination change arising from cloud cover, for example; and small amplitude, local motion of non-stationary background scenery such as foliage or water. The background subtraction process may be inherently robust to these sorts of changes. Since the scope of uninteresting changes is far wider than that of the type we wish to detect, it is difficult for any one method to be robust to all of them. Consequently, separate pre-processing or post-processing stages may be required to handle some of them.

1.1 Applications for moving object detection

Background subtraction is generally considered a lower level image processing task. The segmentation result of the background subtraction stage is then fed into some higher level application, which aims to understand something about the scene. One such application is that of identifying and classifying moving objects based on their appearance. For reliable classification, the object must be described in a compact and robust fashion. The description may include its color distribution, size, shape, and textural information. In order for the description to be characteristic of the true object, a reliable segmentation must be provided. Otherwise, errors in the detection stage will give rise to misrepresentation, which may result in misclassification.

Understanding surveilled *activity* encompasses an array of interesting and challenging problems. Consider the task of object motion tracking, whereby we wish to determine the

position of the same object across time. When there are multiple objects present in the scene, there must be some way of ensuring that tracks correspond to the same object. As in the object recognition problem, an appearance descriptor may be used to measure the similarity of segmented objects in different frames. Again, reliable detections will increase the confidence in matches. Another set of activity understanding problems are human pose and gait recognition, i.e., identifying and distinguishing between a person's orientation like sitting, standing, or crouched, and a person's movements like running, jogging, or walking. These types of tasks require not only good initial object detection, but reliable body part segmentation as well.

Most of these higher level algorithms are designed so that they are robust to a certain level of error in the detection masks. Those which are more sensitive will require image pre-processing or some kind of post-processing of the masks themselves, as is commonly seen. Clearly, improving the background subtraction task itself must transitively improve the higher level applications which it serves.

1.2 Goals of this work

In most of the background subtraction literature to-date, the methods described utilize some kind of pixel-based probability assignment. For this work, the primary method for modeling the background is via the method proposed in (Elgammal et al., 2000). This method was chosen because of it's accuracy, generality, and simplicity. The main contributions presented in this thesis are within the realm of background detection and they apply generally to many existing background modeling methods.

Our goals in this research were to improve the accuracy of the background subtraction method described in (Elgammal et al., 2000). While the authors of that work do describe a post-processing stage to improve the detection results, the method is poorly presented and it is formulated in a seemingly *ad-hoc* fashion. We aimed to find solutions that apply more generally. We have done this by recasting the problem in the classical binary hypothesis testing framework. In doing this, the detection modality, which has traditionally been a

simple fixed thresholding, may be refined by intelligently modeling other image statistics. In addition, by transforming the background modeling process into the statistical *significance* domain, we are able to achieve bounded error rates. This alleviates the algorithm designer from having to choose a threshold heuristically.

These new detection strategies become adaptive to different aspects of the scene. They do not, however, ignore the background statistics as morphological post processing does. That is to say, any strong evidence in the background data to support the assignment of a particular label will not be suppressed unless it is offset by stronger evidence elsewhere supporting the opposite assignment. Many of the commonly used post-processing methods that aim to improve detection results, such as morphological operations, cannot make this claim.

In addition, we require that our methods be robust to slowly changing scenes, i.e., the background model must update over time. We are also particularly interested in robustness to camera shake, which is a common difficulty encountered with outdoor surveillance cameras. As with any processing algorithm, it would be to our benefit to have methods that are relatively simple to implement and light on memory. While these are not strict requirements, details on the algorithm's implementation are noteworthy.

1.3 Thesis overview

The remainder of this thesis is organized as follows. A review of current background subtraction techniques is provided in Chapter 2. Next, some preliminary mathematical concepts that are utilized throughout the thesis are described in Chapter 3. Chapter 4 describes in detail the manner in which we model the background. In particular, three different models are presented and compared. Following the discussion of background modeling methods, Chapter 5 turns to background detection. Two spatially adaptive thresholding strategies are presented which help to suppress false detections and improve the results. The first, refines the decision process by explicitly modeling the foreground in addition to the background. The second strategy, posed as an inverse problem, relies on

modeling the detection mask as a Markov random field. In Chapter 6, we apply a multiple comparison procedure (MCP) known as *false discovery rate control*, which is common in the bio-statistical communities, to the task of background subtraction. The result is a thresholding strategy that not only ensures bounded error rates, but also adapts to the density of moving objects over time. Finally, Chapter 7 offers concluding remarks and points out areas for future research.

Chapter 2

A review of prior work

Moving object detection algorithms for video have existed for some time. Despite the years of work on this topic, research still continues because it is very difficult to obtain reliable detection masks. Also, considering the huge space of all possible scenes and the variety of challenges they may present - global illumination change, camera jitter, camouflaging, small local motion of background structures, etc. - it is no wonder that a single, universally applicable method has yet to be found. In this chapter, we survey the state of the art in background subtraction to provide context for the subsequent presentation of our methods.

2.1 Literal background subtraction

In the simplest of interpretations, the task of background subtraction entails estimating an actual background image, B, and subtracting it from the current video frame I. The absolute value of the difference can be thresholded to find the detection mask, e. The decision at each pixel, $\boldsymbol{n}$ becomes

$$|I[\boldsymbol{n}] - B[\boldsymbol{n}]| \underset{\mathcal{B}}{\overset{\mathcal{F}}{\gtrless}} \phi. \tag{2.1}$$

The threshold can be tuned to account for camera noise, which is often modeled as additive white Gaussian noise. When the scene is truly stationary and the only variation arises from camera noise and moving objects, this simple method of subtracting the background can perform rather well. Accurate estimation and maintenance of the background image is crucial for this method to work, however. Simply taking a single training frame free of moving objects as the estimate for B is generally not sufficient. Firstly, the single frame is

the true B plus the realization of some random, zero-mean noise process, so the estimate is poor. Also, using only a single frame does not allow for any updating.

The 1-D temporal average will provide a good estimate of B when a training sequence (a segment of video with no object motion) is available. Outliers corresponding to foreground objects will corrupt this estimate. This immediately poses a problem when one wishes to update the estimate on-line. A selective filter which updates the background image with only stationary regions of the observed scene may be an attractive alternative:

$$B[\boldsymbol{n}, k+1] = \begin{cases} \rho I[\boldsymbol{n}, k] + (1-\rho) B[\boldsymbol{n}, k] & \text{if } e[\boldsymbol{n}] = \mathcal{B} \\ B[\boldsymbol{n}, k] & \text{if } e[\boldsymbol{n}] = \mathcal{F} \end{cases} \tag{2.2}$$

where $e[\boldsymbol{n}]$ denotes the current detection mask at pixel $\boldsymbol{n}$ and the parameter ρ controls the update speed: ρ of about 10% is usually reasonable. Notice that the top line of (2.2) is a simple temporal auto-regressive estimate, as opposed to the moving average. This representation of the background is recursive, meaning that all previous images are embedded in B, not just recent ones. This can potentially be harmful if the background scene changes significantly. Introducing a certain amount of 'forgetfulness' in the model by increasing the update speed, ρ, can allow the model to adapt to such changes. From a filtering viewpoint, increasing the parameter ρ will shorten the effective impulse response length. For reference, $\rho = 10\%$ has a "memory" of about 44 frames after which there is no significant contribution (less than 0.1%) from previous frames; $\rho = 5\%$ will "remember" about 77 frames.

Alternatively, a non-recursive, or sliding window, approach may be taken which considers only recent frames. Implicitly, a non-recursive approach incurs a memory cost to the algorithm since a frame buffer is required. In place of the temporal mean, which is susceptible to outliers, the temporal median is commonly used. In (Cutler and Davis, 1998), the median across 50-200 recent frames is used to estimate the background image, with the assumption that over this span of time, a pixel is occupied by background at least 50% of the time. Alternatively, in (Jodoin et al., 2006a) $B[\boldsymbol{n}]$ is computed by taking a temporal

median across only 5 frames, spaced apart in time. This is advantageous because the size of the buffer is much smaller. Also, since the frames are non-consecutive, it is unlikely that a non-background outlier will appear in the same location in multiple frames so the median is still a robust estimate.

A recursive median estimate is used in (McFarlane and Schofield, 1995); the running estimate of the median is incremented by one if the current observation is greater than the previous estimate, and decremented by one if it is less. Each pixel in B then eventually converges to a value such that 50% of observed values are above it and 50% are below - i.e., the median. This clearly has the advantage of being both very simple and memory light, since no buffer is required. The detection performance with this method, however, is poor compared to the sliding window median.

In (Cucchiara et al., 2003), an object-based approach is taken whereby moving objects, so-called "ghosts", and shadows thereof are detected based on a number of features of detected foreground regions (size, optical flow, intensity, etc.). The background model is first estimated with a temporal median and then it is updated with knowledge of previous object segmentations, similar to (2.2). This can be thought of as a hybrid between a video *processing* and an *understanding* algorithm whereby information from the higher level of classification is fed back into the lower level task of background subtraction.

2.2 Pixel-based probabilistic models

Although simple subtraction of a background frame will work when the scene is well behaved, generally, this is not the case. Consider, for instance, when there is small local motion in the scene, due to camera jitter or from non-stationary background structures such as foliage, water, or rain. In these more difficult situations, a single image cannot sufficiently describe the complex behavior of the scene. It is more common to model the background with a probability density function (PDF) at each pixel in cases like these. Such a probabilistic approach is more general and is able to better characterize complex scene behavior.

Literal background subtraction, according to Equation (2.1), can be thought of as modeling each pixel with a Gaussian PDF - essentially, the PDF of the noise process shifted by some amount, $B[\boldsymbol{n}]$. Consider a case when a region of the background is not entirely stationary, i.e., there is some small local motion. A single pixel will take on a range of intensities (colors) over time, thus a more accurate probability model would be multimodal.

Many of the pixel based PDF methods in the literature are of the mixture-of-Gaussians (MoG) variety; each pixel's intensity is described by a mixture of K Gaussian distributions where K is usually a small number. The MoG model can be described by the parameters for each component, the mean μ_k, the variance σ_k^2, and a weight w_k. These parameters may be estimated using an expectation maximization (EM) algorithm with recently observed data. This is rather costly computationally and a recursive approach is commonly taken instead. For example, an incremental EM approach is taken in (Friedman and Russell, 1997) whereby the MoG parameters are updated at each time instance. This poses a slight problem, however, since it is not clear how to initialize the model or how to choose an appropriate number of mixture components to begin with.

An on-line K-means approach is taken by (Stauffer and Grimson, 2000). In this approach, each incoming pixel is matched to a cluster if it is within 2.5 standard deviations from the cluster mean. The parameters for that cluster are then updated with the new observation. Repeated similar pixels will drive the weight of their cluster up and simultaneously reduce the cluster variance, indicating a higher selectivity. If no match is found, the observation becomes a new cluster with an initially low weight and wide variance. The quantity w_k/σ_k^2 is in essence a measure of the confidence that a given cluster is background. The clusters which are included in the background PDF are then chosen by sorting each cluster according to w_k/σ_k^2 and selecting the first M of them:

$$M = \underset{m}{arg\min} \left(\sum_{k=1}^{m} w_k > T \right).$$

with T being some predetermined proportion of the clusters. When T is small, $M = 1$ and

the resulting PDFs will be unimodal. When T is allowed to increase, $M \geq 1$ can give rise to multimodal PDFs. Note that the weights of the clusters in the K-means algorithm are not the same as the weights of the MoG mixture components. The M MoG weights must be renormalized so that they sum to 1.

Even this lighter, on-line technique requires a good deal of processing and memory, since the above process must be done for each pixel and must be reiterated for every frame. To speed up this algorithm, a fast variant dubbed the "group of clusters" method is presented in (Butler et al., 2003). In this variant, the clusters are parameterized by only the weight w_k and the centroid c_k, which can each be updated quickly. An incoming observation, x, is matched to the cluster whose centroid that is closest by Manhattan distance; the matched weight is labeled w^*. Background clusters will have large weights and outliers corresponding to foreground objects will fall in smaller clusters. The probabilities that the current observation is either background or foreground are estimated as

$$\Pr\{x \text{ belongs to } \mathcal{B}\} = \sum_{w_k \leq w^*} w_k$$

$$\Pr\{x \text{ belongs to } \mathcal{F}\} = \sum_{w_k > w^*} w_k \, .$$

While fast, this pixel-based probability estimate is coarse and the authors concede that post-processing is necessary to fix misclassifications.

As an alternative to the MoG PDF, the authors of (Elgammal et al., 2000) present a non-parametric approach. Instead of modeling each background pixel by a Gaussian mixture model, a PDF estimate is found using a Gaussian kernel function applied to a recent history of observations at each pixel. Essentially, the PDF estimate is a normalized histogram of the previous observations, filtered by the Gaussian kernel. This approach is attractive for a number of reasons. Firstly, the non-parametric density can be unimodal or multimodal thus it adequately models stationary and dynamic background areas of the scene. In contrast to the MoG approach, there is no ambiguity as to the appropriate number of modes and there are no parameters that need to be estimated on-line. Moreover,

determining the probability of an observation according to the non-parametric PDF can be computed very quickly. The main drawback of this method compared to the methods of (Friedman and Russell, 1997) and (Stauffer and Grimson, 2000) is that it is non-recursive and so it requires a frame buffer. Less this minor drawback, for the same modeling ability of MoG, the kernel-based method is far simpler and faster.

Although there are many more methods which fall under the pixel-based probabilistic heading, we highlight one more which is of particular interest. The aforementioned on-line MoG and kernel-based methods inherently adapt the pixel PDF in time. In (Jodoin et al., 2006a), a framework based on *spatial* distributions is proposed. There, the principle of ergodicity is exploited. The main assumption is that when small local motion is present in the background scene, the statistics of pixel content observed in a small spatial region will resemble those observed over time. Instead of training and maintaining the background PDF with incoming observations, local content from a background image B serves to characterize the background. This background model results in an inherent robustness to small local motions due to camera jitter and textured scenes. The authors apply this spatial framework with both the MoG and kernel methods previously mentioned.

2.3 Other methods

Some other methods, which are not intrinsically pixel-based PDF models, have been presented in the literature.

Similar in principle to the spatial modeling of (Jodoin et al., 2006a), a technique based on a region descriptor dubbed the "local kernel color histogram" is proposed in (Noriega et al., 2006). In that paper, an image is described by color histograms that are smoothed by a Gaussian kernel, much like the non-parametric approach. Unlike (Jodoin et al., 2006a), however, a spatial kernel is also used which ascribes greater influence to nearer pixels. Square 12×12 regions of the image, which overlap, are each described by a histogram. The authors propose applying this region descriptor to the task of background subtraction. Histograms are calculated for both a background image B and the current image I. A

local distance measure is computed as the Bhattacharyya distance between corresponding histograms. That is, the distance

$$d = \sum_k \sqrt{h_k^B \cdot h_k^I}$$

is computed for each pair of histograms (h^B, h^I) in the images.

Since the regions are overlapping, a pixel will fall into multiple histograms. The pixel is therefore assigned a probability based on the spatial kernel and the computed distances of the histograms to which it belongs. This pixel probability is thresholded to obtain a change detection mask. Although a probability is assigned to each pixel, this method is intrinsically region-descriptor based. This implies that the detections will be largely insensitive to small local motions. It also means that the change detection masks will be very smooth and will lack detail, which is evident from the reported results. Additionally, since many local histograms need to be computed and stored as well as the corresponding Bhattacharyya distances, this method will be slow and require a large amount of memory.

Another interesting approach to background subtraction is through the calculation of "eigenbackgrounds" as in (Oliver et al., 2000). Each of N training images (i.e., containing only background) are represented as vectors: if the images are natively $(H \times W)$ in dimension, the vector representation is $(HW \times 1)$. The mean of each pixel and the covariance matrix is determined from these N frames. An eigenvector decomposition of this, very large $(HW \times HW)$ covariance matrix is performed and M of the vectors with the largest eigenvalues, also known as the principle components, are kept in a matrix Φ_M. The M vectors in Φ_M correspond to the so-called "eigen"-background images; their mean denoted μ_b. Incoming images are mean subtracted and projected onto this low dimensional subspace. The residual reconstruction error, or "difference from feature space" (DFFS), defined as

$$\epsilon^2(I) = ||I - \mu_b||^2 - \sum_{m=1}^{M} \Phi_M^T (I - \mu_b)$$

is thresholded to detect motion (Moghaddam and Pentland, 1997). The authors of (Oliver

et al., 2000) cite that the eigenbackground approach has lighter computational load than an MoG type of description. However, they also rely on a morphological post processing to produce their final blob segmentations of pedestrians before higher level processing is performed.

The "Wallflower" algorithm, designed for body part segmentation and gesture recognition for is presented in (Toyama et al., 1999). The initial phase of background subtraction in that work was based on a temporal Wiener filter at each pixel. Based on 50 previous observations, a linear prediction of the current observation is computed. Using a history of observations, the Wiener filter is able to capture periodic time-varying behavior of the background under Gaussian noise. The difference between the prediction and the actual observation is thresholded to detect foreground objects. Foreground objects themselves can corrupt the recent history, however. The solution offered by the authors is to apply a *second* Wiener filter to, not the observed values, but the predicted values of the first Wiener filter. This clearly doubles the required computation.

Chapter 3

Preliminaries

Before delving into the background subtraction methods of this paper, it is necessary to introduce some concepts and notation which will be used throughout.

In this paper our presentation is based on grayscale intensity image sequences denoted $I[\boldsymbol{n}, k]$. The second index, k denoting the time or frame number in the sequence, is used minimally here and will often be omitted in the text to simplify notation. The spatial index is indicated by the vector (boldface) $\boldsymbol{n} = [n_1\ n_2]^T$ and exists on an orthogonal lattice:

$$\boldsymbol{n} \in \Lambda \in \mathbb{R}^2$$

Intensity values are quantized to integers in the range $[0, 255]$. Although we consider only grayscale, extensions to higher spaces of color are straightforward.

The detection result of background subtraction is a binary detection mask, or label field. Throughout this thesis, the detection mask at pixel $\boldsymbol{n}$ shall be denoted with the symbol $e[\boldsymbol{n}]$: $e[\boldsymbol{n}] = 0$ if pixel $\boldsymbol{n}$ is determined to arise from the background and $e[\boldsymbol{n}] = 1$ if it belongs to the foreground.

3.1 Binary hypothesis testing

Performing background subtraction on an image $I[\boldsymbol{n}]$ entails applying a label, either $\mathcal{B}$ or $\mathcal{F}$, to each pixel by comparing the current image to some background model. The problem can be formulated as hypothesis testing (Theodoridis and Koutroubas, 2006). In

the common fashion, the decision is framed thusly:

$$\frac{\Pr\left\{\mathcal{I}[\boldsymbol{n}] = I[\boldsymbol{n}] \;\middle|\; \mathcal{B}\right\}}{\Pr\left\{\mathcal{I}[\boldsymbol{n}] = I[\boldsymbol{n}] \;\middle|\; \mathcal{F}\right\}} \underset{\mathcal{F}}{\overset{\mathcal{B}}{\gtrless}} \eta \frac{\pi_{\mathcal{F}}}{\pi_{\mathcal{B}}}. \tag{3.1}$$

The probabilities on the left hand side denote the probability of observing the intensity value $I[\boldsymbol{n}]$ (the realization of the random field $\mathcal{I}[\boldsymbol{n}]$) given that it is the projection of either the background scene or a foreground object. To simplify notation, the probability $\Pr\left\{\mathcal{I}[\boldsymbol{n}] = I[\boldsymbol{n}] \;\middle|\; \mathcal{B}\right\}$ will henceforward be denoted more compactly as $P_{\mathcal{B}}(I[\boldsymbol{n}])$, and likewise for $P_{\mathcal{F}}(I[\boldsymbol{n}])$. The functions $P_{\mathcal{B}}$ and $P_{\mathcal{F}}$ are probability density functions (PDFs) which shall be discussed in more detail in Section 3.3. The ratio of these probabilities is known as the *likelihood ratio* and the (3.1) is called a likelihood ratio test (LRT). The right hand side contains the prior probabilities of observing background $\pi_{\mathcal{B}}$ or foreground $\pi_{\mathcal{F}}$ and a cost term η. The prior ratio biases the decision based on the the *a priori* probability of observing each outcome. The cost term can be designed to incur unequal penalty to the four decision/truth scenarios: $\mathcal{B}/\mathcal{B}$, $\mathcal{B}/\mathcal{F}$, $\mathcal{F}/\mathcal{F}$, and $\mathcal{F}/\mathcal{B}$.

3.2 Multiple comparisons procedures

In a binary hypothesis test, there is one test which results in one of two possible outcomes. This deterministic decision rule maps an observation onto the 2-element decision space, commonly denoted by $\{H_0, H_1\}$ corresponding to the null and positive hypotheses respectively; equivalently $\{\mathcal{B}, \mathcal{F}\}$ in our treatment. In the context of background subtraction, this test is being performed at every pixel in an image, thus we have many such binary tests.

The framework of so-called *multiple comparisons procedures* (MCP) is essentially the same; there are M binary tests each with a unique probability distribution under the null hypothesis. A decision rule maps each of M observations into a decision of either H_0 or H_1 ($\mathcal{B}$ or $\mathcal{F}$). To facilitate further discussion, we now introduce some new terminology relevant to MCP. Table 3.1 illustrates the definitions of random variables which denote *quantities* of trials; that is, each takes an integer value in the range $[0, M]$. In general, U,

V, T, and S are unobservable random variables while R is observable (as is the quantity $M - \mathsf{R}$, naturally).

Table 3.1: Definition of random variables for MCP.

	# declared H_0	# declared H_1
# true H_0	U	V
# true H_1	T	S
Total	$M - \mathsf{R}$	R

With these random variables defined, we may now define error rates which are of particular importance. The global false positive rate (FPR) and false negative rate (FNR) are defined thusly:

$$FPR = \mathbb{E}\left\{\mathsf{V}\right\}/M \tag{3.2}$$

$$FNR = \mathbb{E}\left\{\mathsf{T}\right\}/M \tag{3.3}$$

where $\mathbb{E}\left\{\cdot\right\}$ denotes the expectation operator. Additionally, the total error rate (TER) is simply the sum of these two individual error types:

$$TER = \mathbb{E}\left\{\mathsf{V} + \mathsf{T}\right\}/M. \tag{3.4}$$

For our purposes, these quantities suffice to qualify a particular detection strategy and of course our general aim is to reduce all error rates. In addition, the so-called *false discovery rate* becomes a quantity of interest in Chapter 6. It is defined as the expected proportion of H_1 declarations which are erroneous:

$$FDR = \mathbb{E}\left\{\mathsf{V}/\mathsf{R}\right\}. \tag{3.5}$$

When the ground truth is known, the realizations of all quantities in Table 3.1 are observable. The error rates can be estimated by averaging observed quantities v, t, and r over multiple trials.

3.3 Non-parametric PDF estimate

The background modeling stage consists of characterizing the typical intensity values that each pixel takes. In other words, a probability density function is estimated at each location. As discussed in Chapter 2, there are a number of ways that this can be done. Following the example of (Elgammal et al., 2000; Elgammal et al., 2002), this work considers models of the non-parametric type. The non-parametric estimate is attractive because it converges to the true underlying PDF with an appropriately selected kernel function (Theodoridis and Koutroubas, 2006). Let the kernel functions, $\mathcal{K}$, be zero-mean Gaussians parameterized by variance σ^2:

$$\mathcal{K}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{x^2}{2\sigma^2}\right)$$

A technique for adaptively estimating the kernel bandwidth (variance) at each pixel is presented in (Elgammal et al., 2000; Elgammal et al., 2002). In this work, however, a constant kernel variance for the entire image is used for simplicity and ease of computation. This is sufficient because any observed variability in the intensity that is not caused by movement or change in the scene itself arises from camera noise. Our assumption is that the camera noise is consistent across the entire image. The PDF is constructed from sample data y by summing shifted copies of the kernel:

$$P(x) = \frac{1}{|\mathcal{M}|} \sum_{y \in \mathcal{M}} \mathcal{K}(x - y). \tag{3.6}$$

The set $\mathcal{M}$, denoting the particular model, is comprised of intensity values y. The manner in which pixels are added to the model will be discussed in Chapter 4.

The intensity values (x, y, and $I[n]$) are discretized to integers in the range $[0, 255]$. $P(x)$ may alternatively be viewed as the histogram of $y \in \mathcal{M}$, normalized by $|\mathcal{M}|$ and filtered by $\mathcal{K}(x)$.

The following examples serve to clarify the point. We have generated 100 random observations (black x's) from two known underlying PDF's (blue curves), one unimodal

and one multimodal. The non-parametric density estimate built from the observed data is shown (green curves).

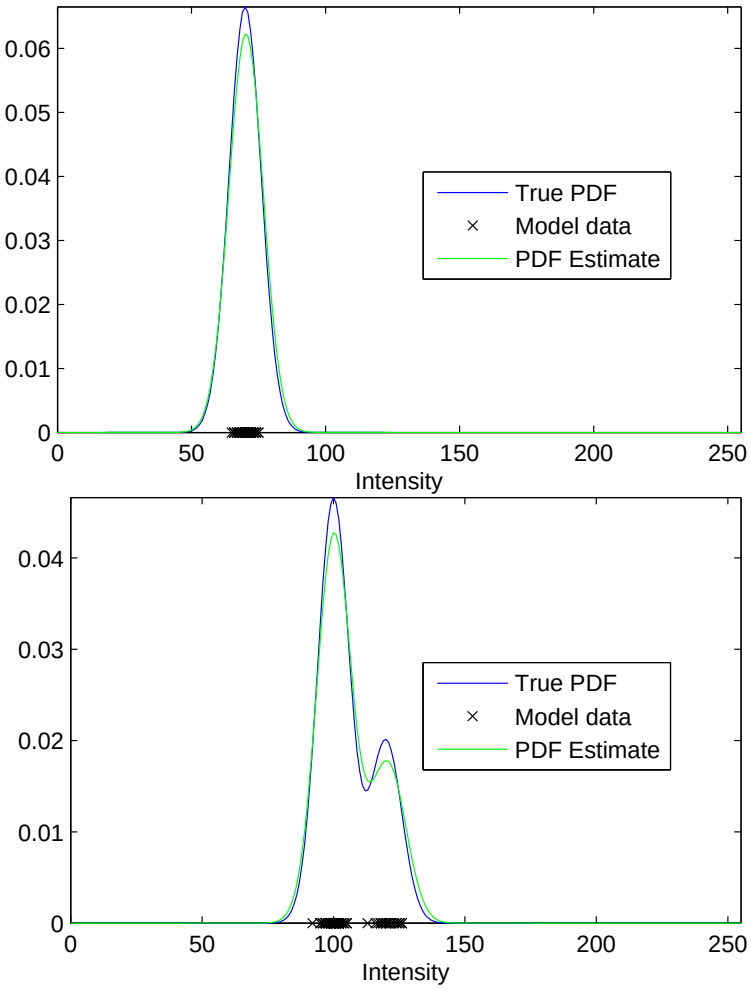

Figure 3.1: Examples of two non-parametric PDF estimates. The true underlying PDF is shown in blue. Samples generated from this distribution are denoted with black x's and the non-parametric density estimate built from these samples is shown in green.

From an implementation perspective, computing an entire density function at each pixel is costly and requires a significant amount of storage. Thankfully, it is also often unnecessary. Since we only wish to classify the current observation as background or foreground, we only need the probability of the current observation according to the given distribution: the value $P_{\mathcal{B}}(I[\boldsymbol{n}])$. This can be exploited when implementing the modeling process.

Since good background models are adaptive, the PDF is constantly changing. Regularly updating the entire PDF (i.e., all 256 values) for every pixel can be very computationally intensive. Even when the PDF updating scheme is intelligent - discard only old samples and incorporate only new samples as opposed to rebuilding from scratch each time - computing a single value has a large advantage over computing 256 values. With x replaced by $I[\boldsymbol{n}]$ in (3.6), the summation can be viewed as aggregating the similarity of each model pixel y with the current observation. Since the kernel function is symmetric and decreasing in the *absolute value* of the argument, small absolute differences contribute largely to the similarity measure, while large absolute differences will only contribute slightly. Furthermore, since the density functions are discretized, the values of $\mathcal{K}(|x|)$ can be precomputed then looked up from a list at run time.

3.4 Simplified background detection

In the forthcoming chapter, several background modeling methods are presented and compared. To frame the discussion as *background subtraction*, a particular *detection* modality must be defined. Within the framework of (3.1), a simple detection method is herein derived. Later, in Chapters 5 and 6, the detection process will become more sophisticated.

In many hypothesis testing problems, it is common to translate the decision rule of (3.1) into the observation space. Here, it is more convenient to stay in the probability space. The simplest background detection process involves comparing a value of $P_{\mathcal{B}}(I[\boldsymbol{n}])$ to some threshold, θ. A pixel is thereby labeled as $\mathcal{F}$ if the probability is sufficiently small, i.e., the observed value is unlikely to have come from the background. Consider this in terms of the framework of (3.1). The probability of observing a particular intensity value $I[\boldsymbol{n}]$ when the pixel is occupied by a moving object is uncertain. Since a moving object's presence in the scene is transient, it is appropriate to assume that it's intensity (color) distribution is unknown. $P_{\mathcal{F}}(I[\boldsymbol{n}])$ is therefore considered to be uniformly distributed over all intensity values. Taking on a constant value, $P_{\mathcal{F}}(I[\boldsymbol{n}])$ can be collected with the other

constants, $\pi_{\mathcal{B}}$,$\pi_{\mathcal{F}}$, and η, on the right-hand side to arrive at a fixed threshold test:

$$\boxed{P_{\mathcal{B}}(I[\boldsymbol{n}]) \underset{\mathcal{F}}{\overset{\mathcal{B}}{\gtrless}} \theta} \tag{3.7}$$

Chapter 4

Background modeling

We now turn our attention to the background PDF $P_{\mathcal{B}}$. Here I define three different background models which may all be described by the non-parametric PDF discussed in Section 3.3. This method of modeling the background is advantageous because it can describe complex scenes that are not completely static. Also, the implementation is very fast and straightforward since determining the value $P_{\mathcal{B}}(I[\boldsymbol{n}])$ requires only additions and list lookups (since the intensity data are quantized to integers). Also, unlike an MoG background model, there are no parameters that need to be estimated.

Each of the three models presented in this chapter differ in the composition of the set of model pixels, $\mathcal{M}$. Therefore, the three models will be referred to as $\mathcal{M}_i$ with $i = 0, 1, 2$. The 'zeroth' model, so dubbed because it is a degenerate case, corresponds to a literal subtraction of a background image from the current frame. The next model is comprised of a set of recently observed pixels. This was originally proposed in (Elgammal et al., 2000) and is the basis for much of this work. The third model considered in this thesis was originally proposed in (Jodoin et al., 2006b) and is based on pixels observed locally in space, rather than pixels observed locally in time.

These three background models provide different ways to describe a particular scene. As with many engineering problems, there is not necessarily one universally *best* method. Over a range of scenarios, tradeoffs must be made between model accuracy, memory requirements, and processing time. These issues become pertinent when we evaluate the detection methods presented in Chapter 5.

4.1 Background frame: $\mathcal{M}_0$

4.1.1 Definition

In the most literal interpretation of "background subtraction" a background image $B[\boldsymbol{n}]$ is pixel-wise subtracted from the current frame and the absolute difference is compared to a threshold. This can be thought of as modeling the pixel by a Gaussian PDF with mean $B[\boldsymbol{n}]$ and variance σ^2 which accounts for camera noise. The similarity is illustrated in Fig. 4.1. A threshold in terms of the probability, θ, transforms directly to a threshold in terms of the absolute difference, ϕ, via the Gaussian function:

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \underset{\mathcal{F}}{\overset{\mathcal{B}}{\gtrless}} \theta$$

$$|x-\mu| \underset{\mathcal{B}}{\overset{\mathcal{F}}{\gtrless}} \phi \equiv \sqrt{-\sigma^2 \ln\left(2\pi\sigma^2\theta^2\right)}$$

In terms of the non-parametric density of (3.6), the single Gaussian model is a degenerate case. Nevertheless, let the model for pixel $I[\boldsymbol{n}, k]$ be defined as

$$\mathcal{M}_0(I[\boldsymbol{n}, k]) = \left\{ y \;\middle|\; y = B[\boldsymbol{n}] \right\}. \tag{4.1}$$

Thresholding the intensity difference is more straightforward in terms implementation. However, when this simple method is abstracted out to a probabilistic framework, it aligns nicely with the other models discussed in the following sections.

4.1.2 Issues

As discussed in Chapter 2, there are a number of ways in which the background frame may be updated. One may recursively update B by blending non-detected regions of incoming frames, or take a temporal mean or median over a set of buffered frames. These methods fail however when the camera jitters, causing frames to become misaligned, as shown in Fig. 4.2. Currently, wind load causes our outdoor surveillance cameras to sway and shake, sometimes violently. To combat camera shake, frames are preregistered with a technique called phase correlation, or PC. The method itself is described in detail in Appendix A.

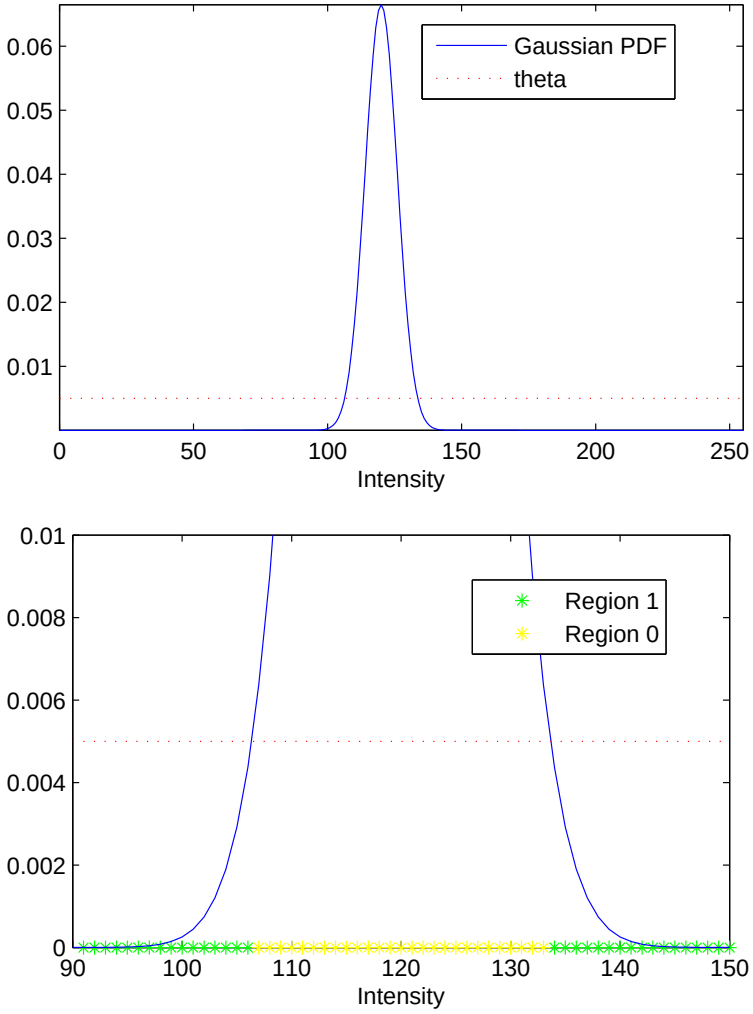

Figure 4.1: Probabilistic interpretation of literal background subtraction. When the background PDF is symmetric, e.g., a single Gaussian ($\mu = 120$ and $\sigma^2 = 6$ here), comparison to a threshold θ yields decision regions equivalent to those achieved by thresholding the absolute difference $|x - \mu|$.

Even when frames are preregistered using PC, small residual misalignments will cause false detections along high contrast edges, as can be seen in Fig. 4.3.

Another difficult case for literal background subtraction is a textured background that moves slightly, e.g., foliage or water. In these cases, a background pixel will commonly take on a range of intensity values which the simple model of M_0 cannot capture. In parts of the scene that are static, like the highway in Fig. 4.4, the simple background frame model performs rather well. When the background is not entirely stationary however, literally

Figure 4.2: When the camera shakes, simply subtracting the background is clearly insufficient. The background frame itself (top) is distorted as a result of ρ-blending misaligned frames together.

subtracting a background image is insufficient. To deal with the difficulties discussed, more sophisticated models must be employed.

4.2 Local-in-time model: $\mathcal{M}_1$

4.2.1 Definition

The background model denoted $\mathcal{M}_1$ is constructed in the manner described in (Elgammal et al., 2000; Elgammal et al., 2002). At each location $\boldsymbol{n}$, the model consists of a recent history of N intensity values at that location.

$$\begin{aligned} \mathcal{M}_1(I[\boldsymbol{n}, k]) &= \left\{ y \;\middle|\; y = I[\boldsymbol{n}, k-l] \right\} \\ l &= 1, 2, \ldots, N \end{aligned} \tag{4.2}$$

The number of recent frames, N, is usually around 50-100, which corresponds to few seconds of video (the frame rate is around 25 frames per second).

Figure 4.3: Realigning the frames before performing detection substantially improves the detection performance when using $\mathcal{M}_0$ as well as the quality of the background image (top). False positives along high contrast edges are still present however. Details of the realignment technique used can be found in Appendix A.

Figure 4.4: Although the cars on the highway below are well detected, the trees in this frame, whose leaves and shadows thereof displace slightly, are incorrectly detected as moving objects using model $\mathcal{M}_0$.

In Fig. 4.5 the recent history at a single pixel is shown along with the resulting PDF. Notice that outliers in the history (in this case corresponding to moving objects) will cause

the model to characterize non-background. If the number of outliers is small compared to the total number of samples, the effects are minimal. Notice the transient state of the pixel intensity in Fig. 4.5 contributes a non-background peak to the PDF. Clearly, this can lead to misses if another object of similar intensity (color) occupies this pixel in the near future. For this reason, it is wise to *selectively* update the background model, i.e., only add a new pixel to the model if it has previously been declared background. In contrast, the simple updating technique whereby pixels are added to the model regardless of their previous label is termed a *blind update* (Elgammal et al., 2002).

4.2.2 Sample detection results

The local-in-time (LIT) background modeling method has long been favored for its effectiveness and simplicity, and it is the basis for much of this work. The moving nature of this model in time means that it is inherently adaptive to gradual scene changes. Furthermore, it was designed to be able to characterize difficult, non-stationary scenes: e.g., an outdoor scene with a tree in the background whose leaves and branches move with the wind. A pixel that is occupied by more than one object over time (e.g., a leaf, a branch, and the sky) can effectively be modeled by a multimodal distribution using the non-parametric density estimate.

To some extent, the model can also handle slight camera jitter. When the shakiness of the camera is substantial, which it happens to be in our particular case, the model will fail as illustrated in Fig. 4.7. Notice that many errors are present, especially along high contrast edges. When the frames have been aligned with the PC preprocessing step (see Appendix A), the detection is much better. The PC method we use determines a global displacement to within single pixel accuracy. While it is possible to perform subpixel registration with this same method, redrawing a frame with a fractional offset requires interpolation, which has an undesired lowpass smoothing effect. Occasionally, a slight misregistration can cause detection errors, as illustrated in Fig. 4.8. Since the registration is accurate to within one pixel, these specific types of errors tend to manifest

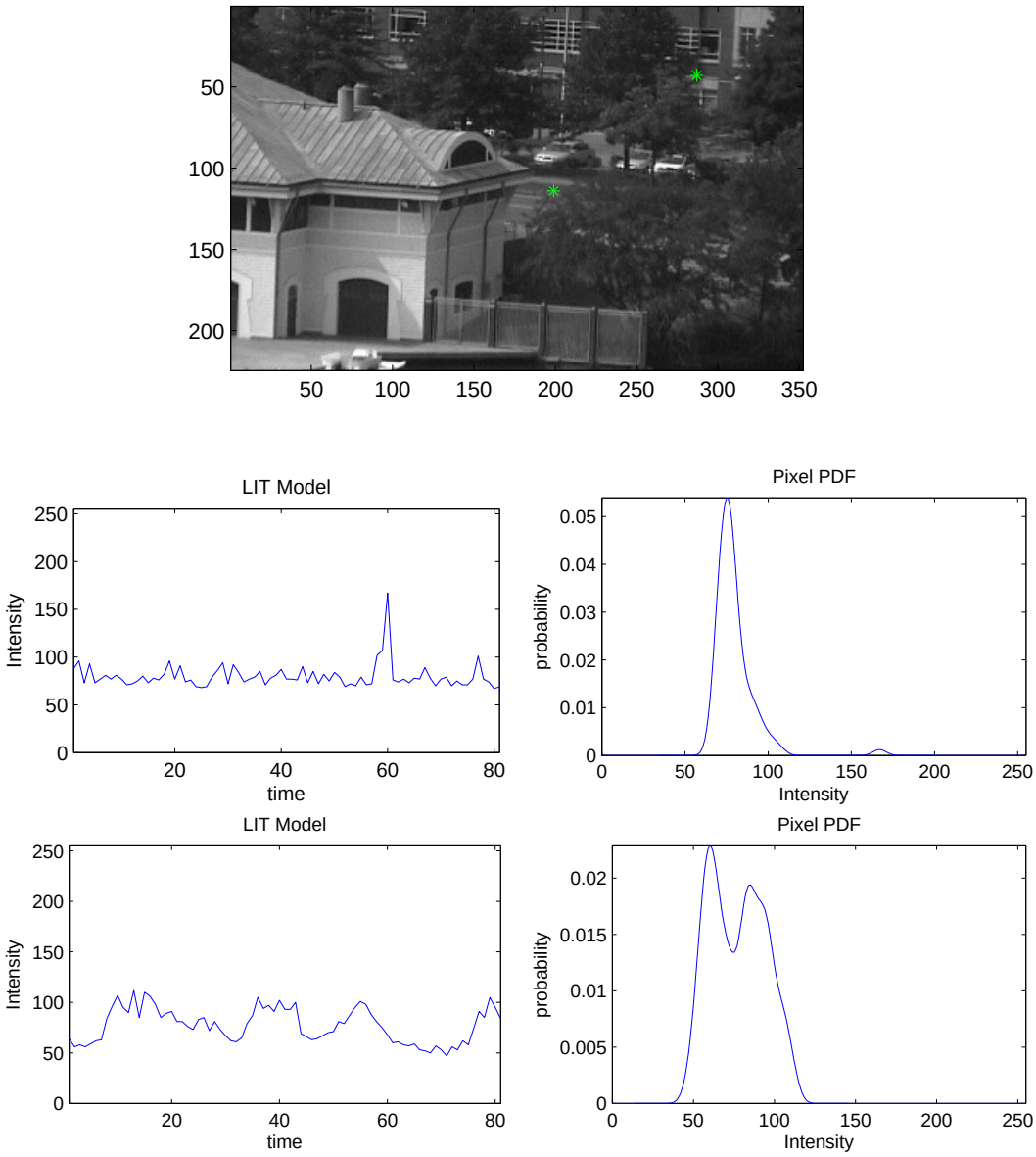

Figure 4.5: Frame from a test sequence with two pixels highlighted. One pixel corresponds to the stationary road (first row of graphs), the other pixel is occupied both a moving branch and the wall of a building (second row of graphs). The pixels' intensity values over a recent time period which comprise $\mathcal{M}_1$ and the probability density functions are shown.

as single-pixel-wide lines along edges.

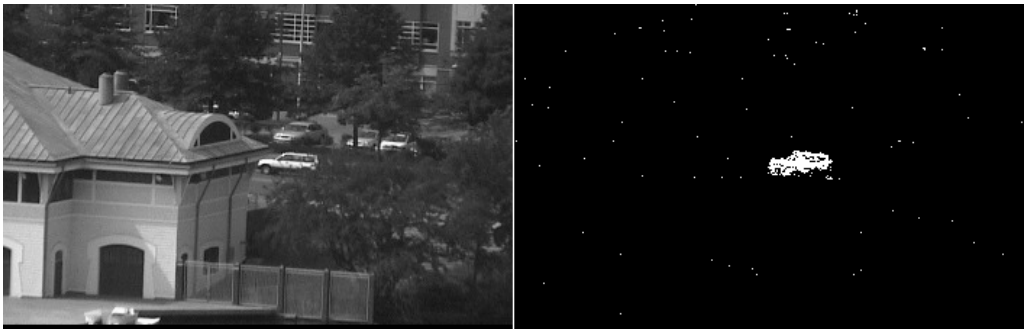

Figure 4.6: The trees in this sequence move slightly with the wind, however $\mathcal{M}_1$ is able to sufficiently model the dynamic background. The resulting detection mask highlights the moving object of interest, the car, and not the unimportant motion of the trees.

Figure 4.7: When camera shake is present, the background subtraction result can have a high number of false detections when using $\mathcal{M}_1$ (left). Performing realignment on the frames before modeling takes place mitigates most of these errors (right). Details of the realignment technique used can be found in Appendix A.

4.3 Local-in-space model: $\mathcal{M}_2$

4.3.1 Definition

The local-in-space (LIS) model was proposed in (Jodoin et al., 2006a; Jodoin et al., 2006b) as a memory-light alternative to LIT type models. Instead of training the background

Figure 4.8: Although image preregistration will greatly improve the detection performance using $\mathcal{M}_1$, occasional misregistration may still result in detection errors along edges.

model on recent frames as for $\mathcal{M}_1$, nearby pixel content from a single background frame is used to characterize the model. The definition for the LIS model is

$$\begin{aligned} \mathcal{M}_2(I[\boldsymbol{n}, k]) &= \left\{ y \;\middle|\; y = B[\boldsymbol{m}] \right\} \\ \boldsymbol{m} &\in \mathcal{N}(\boldsymbol{n}) \end{aligned} \tag{4.3}$$

where $\mathcal{N}(\boldsymbol{n})$ denotes a spatial neighborhood around pixel $\boldsymbol{n}$ (e.g., a 9×9 square region centered at $\boldsymbol{n}$). Figure 4.9 illustrates an example of a PDF built from spatially local content.

This method exploits the principle of ergodicity. That is, the statistics of pixel content observed over time, as in the LIT model, should be similar to pixel content observed in a small spatial region. This is especially true when small local motion is present.

4.3.2 Sample detection results

The first advantage of the LIS model over the LIT version is that it only requires a single frame to characterize the background, meaning that this model is light on memory. Secondly, the LIS model is much less susceptible to errors due to camera shake. Since local pixels are included in the model, small spatial displacements do not manifest as false detections as they would with $\mathcal{M}_1$. In Fig. 4.10, sample detection results are presented on two very shaky image sequences. The commonly seen errors due to camera shake are

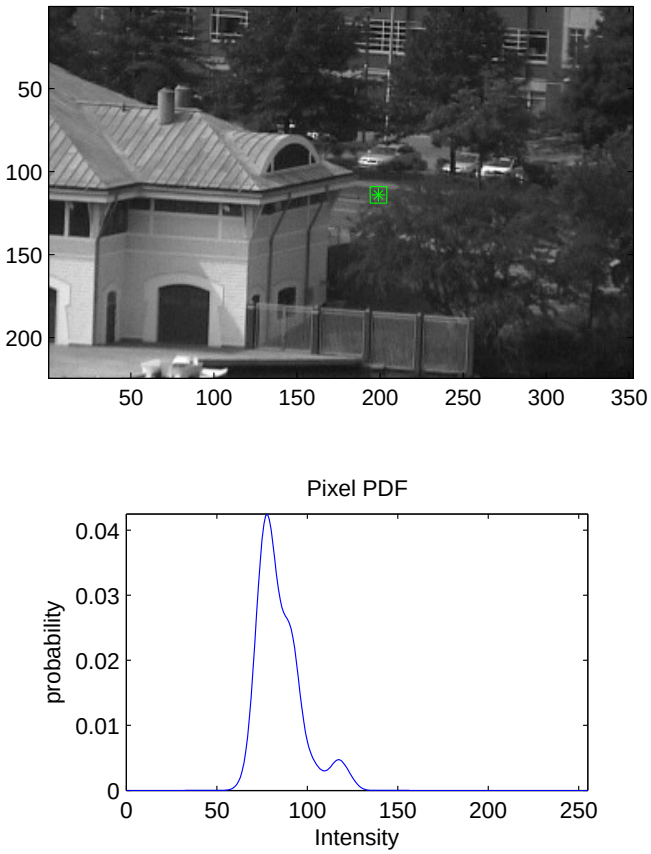

Figure 4.9: Pixels in a small region in the background frame, B, comprise the model $\mathcal{M}_2$ for the pixel at the center of the region. The probability density function built from this model is shown below.

mitigated by using this background model without the need for any preprocessing.

A drawback of using spatially local information to describe a single pixel is that the PDFs will characterize background *regions* and can be wide in inhomogeneous regions. This can result in increased misses due to camouflaging. Notice in Fig. 4.11 the large missed region at the roof of the van. This can be attributed to the similarity of the foreground object with a nearby region of the background.

Figure 4.10: The LIS non-parametric model ($\mathcal{M}_2$) performs well on unregistered images even under substantial camera shake.

Figure 4.11: Describing a pixel by the content of a region can 'over-model' the background in some cases, resulting in large missed foreground regions. A portion of the roof of the van is incorrectly labeled as background with $\mathcal{M}_2$ because it resembles a nearby region of the scene.

4.4 Discussion

Three background modeling methods which apply to a non-parametric PDF estimate have been presented. While the simple subtraction of a background frame works relatively well when the scene is static, small motions are better handled by the LIT and LIS models.

In our particular case, the cameras themselves and scenes of interest are outdoors thus camera shake and dynamic background structures are often present. Comparing the three methods, we've found that LIT with pre-registration is the best solution in most cases. It provides an accurate model of the scene and it does not suffer from unnecessarily over-modeling the background as the LIS model can. In terms of memory requirements, $\mathcal{M}_1$ is the heaviest. The LIT model requires a large frame buffer, whereas the background frame and LIS models are lighter on memory since they require only a single frame to characterize the scene. Model $\mathcal{M}_1$ also requires the pre-processing step to handle camera jitter, which $\mathcal{M}_2$ does not.

One difficulty with all of these techniques is that they are prone to exhibit randomly scattered false positives, even with a low threshold. In addition, residual misalignment from image preregistration can also give rise to false positives, especially along high-contrast edges as we have shown. Again, lowering the global threshold is not a desirable solution since this will increase the miss (false negative) rate. Another very difficult situation is that of camouflaging - when the moving object resembles the modeled background. In the LIT case, this can be the result of outliers in the background history that have been erroneously added to the model. In the LIS case, the camouflage may be due to over modeling the pixel with data from an inhomogeneous region, as we have seen.

In the following chapter, we present methods whereby the detection threshold is *biased* spatially according to new image statistics. Our goal is to discourage both types of errors simultaneously resulting in more reliable and more natural looking detection masks.

Chapter 5

Spatially adaptive background detection

Upon inspection of the detection results presented in Chapter 4, one notices that even with good background modeling there is still room for improvement. In almost all cases, there tend to be false positives scattered randomly across the image. Furthermore, false negatives manifest as incompletely detected objects and jagged object boundaries. In some higher level vision applications when relatively infrequent or inconsistent errors can be averaged out or ignored, this may be acceptable. Applications of behavior modeling or average motion monitoring may fall into this regime. More commonly however, in tasks involving object tracking or identification for example, reliable object detection is paramount. In either case, improving the performance of the background subtraction task will benefit the higher level application.

It is possible to perform post-processing on the detection masks in order to suppress false detections. For example, median filtering can reduce the so called *salt-and-pepper* noise that is characteristic of randomly scattered errors. Morphological processing, such as region growing and shrinking, could also be employed in an effort to fill in missing regions. While these methods can be effective in some cases, they operate only on the detection mask, ignoring the original image statistics. Contrarily, the detection methods described herein remain within the general probabilistic framework described by the LRT (3.1) and aim to intelligently refine the decision process itself. The proposed methods apply generally to any background probability measure $P_{\mathcal{B}}(I[\boldsymbol{n}])$, not only those discussed in Chapter 4.

Recall that in the previous discussion, the background detection modality was comparison of the background probability to some fixed threshold: $P_{\mathcal{B}} \gtrless \theta$. By adjusting the global threshold, θ, performance in terms of error rate will improve for one error type but

will suffer for the other; e.g., by lowering the threshold, the false positive rate will decrease but the miss rate will increase. With this simple fixed-thresholding detection strategy, constancy of the foreground distribution and prior terms is implicit. This thesis describes how these terms can be statistically modeled and exploited in order to improve the detection performance. By adapting the threshold spatially over the image, in accordance with probabilistic models for $\pi_{\mathcal{B}}$, $\pi_{\mathcal{F}}$, and $P_{\mathcal{F}}$, we can selectively discourage both types of errors simultaneously.

In this chapter, two adaptive background detection methods are presented. Each can be viewed as a spatially variable threshold biasing strategy. Section 5.1 describes a method for estimating the *foreground* probability distribution $P_{\mathcal{F}}(I[\boldsymbol{n}])$ from locally detected pixels. Next, a method for controlling the prior weights $\pi_{\mathcal{B}}$ and $\pi_{\mathcal{F}}$ is presented in Section 5.2 whereby the detection mask is modeled as a Markov random field. Comments about combining these methods are offered in Section 5.3. The final section of this chapter provides a numerical performance comparison.

5.1 Explicit foreground object modeling

In the previous chapter and in most background subtraction algorithms to-date, background detection considers only statistics of the background model. As discussed in Section 3.4, the foreground probability density is usually considered to be uniform. By estimating foreground statistics more precisely, one may expect greater discrimination ability.

In some specific applications, such as body part segmentation, foreground object region growing has been done. In (Wren et al., 1997) and in (Elgammal et al., 2002), foreground regions are morphologically grown according to specific shape and color models of the head, torso, and extremities. In both of these works however, the foreground blob modeling is entirely separate from the foreground detection process itself. In the "Wallflower" algorithm presented in (Toyama et al., 1999), foreground regions corresponding to arbitrary objects, not necessarily body parts, are grown based on color histograms. Again, however, the foreground region growing is performed as a secondary step in which the background

statistics are ignored. This means that nearby background content that resembles the initially detected foreground is likely to become erroneously included in the grown foreground blobs.

The method proposed in this thesis is distinct from these previous works in that the foreground statistics are estimated and included in the background subtraction task itself, which includes the already learned background statistics. Also, no prior shape or orientation of the foreground is explicitly assumed in our method. Details of the proposed method, implementation notes, and some experimental results are presented in this section.

5.1.1 Proposed method

Our aim is to replace the naive, uniform foreground PDF model with a more accurate estimate based on the appearance of the actual object. Similar to the formulation for $P_{\mathcal{B}}$ in Chapter 4, the foreground probability $P_{\mathcal{F}}$ is constructed via a Gaussian kernel, non-parametric density estimate. The intensity (color) distribution of foreground objects may be unimodal but it will likely be multimodal. A non-parametric density is therefore appropriate for modeling and arbitrary object and it is desirable over an MoG approach for the same reasons discussed in Chapters 2 and 4.

Since the location of moving objects is time varying, a model consisting of recent pixels at a single location will not suffice for $P_{\mathcal{F}}$. Instead, pixels detected in a small spatial locality in the current frame are used, similar to the LIS background model. The proposed method is detailed as follows.

Let $e[\boldsymbol{n}]$ denote the detection label field at pixel $\boldsymbol{n}$; $e[\boldsymbol{n}] = 0$ if pixel $\boldsymbol{n}$ belongs to the background and $e[\boldsymbol{n}] = 1$ if it belongs to the foreground. Using the naive uniform foreground model and the simplified LRT of (3.7), an initial detection mask is found, denoted $e_0[\boldsymbol{n}]$. Next, we define a set of locally detected pixels as

$$\mathcal{N}_{\mathcal{F}} = \{\boldsymbol{m} \in \mathcal{N}(\boldsymbol{n}) : e_0[\boldsymbol{m}] = 1\}$$

where $\mathcal{N}(\boldsymbol{n})$ denotes a local neighborhood around $\boldsymbol{n}$ - e.g., a 7×7 window centered at

$\boldsymbol{n}$. The foreground PDF is then estimated using the same kernel-based approach as before from pixels of the current frame at the locations specified by $\mathcal{N}_{\mathcal{F}}$.

$$P_{\mathcal{F}}(I[\boldsymbol{n}]) = \frac{1}{|\mathcal{N}_{\mathcal{F}}|} \sum_{\boldsymbol{m} \in \mathcal{N}_{\mathcal{F}}} \mathcal{K}(I[\boldsymbol{n}] - I[\boldsymbol{m}]) \tag{5.1}$$

If there are no detected pixels in the neighborhood of $\boldsymbol{n}$ (resulting in $\mathcal{N}_{\mathcal{F}}(\boldsymbol{n}) = \emptyset$), there is presumably no foreground object at $\boldsymbol{n}$ to model and we revert to the naive assumption that $P_{\mathcal{F}}$ is uniform and takes on a constant value.

Using the stored values of $P_{\mathcal{B}}(I[\boldsymbol{n}])$ and with $P_{\mathcal{F}}(I[\boldsymbol{n}])$ computed at every $\boldsymbol{n}$, the refined likelihood ratio can be retested to obtain a new label field e_1 as follows:

$$\boxed{\frac{P_{\mathcal{B}}(I[\boldsymbol{n}])}{P_{\mathcal{F}}(I[\boldsymbol{n}])} \underset{\mathcal{F}}{\overset{\mathcal{B}}{\gtrless}} \eta} \tag{5.2}$$

Note that the threshold on the right-hand side of (5.2) is related to the original θ of (3.7), which implicitly includes the constant value of the naive uniform PDF

$$\eta \cdot (1/256) = \theta.$$

Since we have assumed nothing *a priori* about the label $e[\boldsymbol{n}]$ thus far, we allow $\pi_{\mathcal{F}}/\pi_{\mathcal{B}} = 1$. That is, both assignments are equally likely.

From (5.2) it can be seen that $P_{\mathcal{F}}(I[\boldsymbol{n}])$ defined for all $\boldsymbol{n}$ acts as a spatially variable threshold biasing term, as we illustrate in Fig. 5.1. If for a particular intensity $I[\boldsymbol{n}]$ its foreground probability $P_{\mathcal{F}}(I[\boldsymbol{n}])$ is lower, then the effective threshold $\eta \cdot P_{\mathcal{F}}(I[\boldsymbol{n}])$ is reduced thus encouraging assignment of the background label. To the contrary, a higher value of $P_{\mathcal{F}}(I[\boldsymbol{n}])$ will encourage assignment of the foreground label.

5.1.2 Notes on implementation

The test (5.2) can be reiterated with the new label field e_1 used to determine $\mathcal{N}_{\mathcal{F}}$ and corresponding $P_{\mathcal{F}}$, and repeated in this fashion. We have noticed, however, that the process is slow to converge. Fortunately, the most significant gains occur within the first few

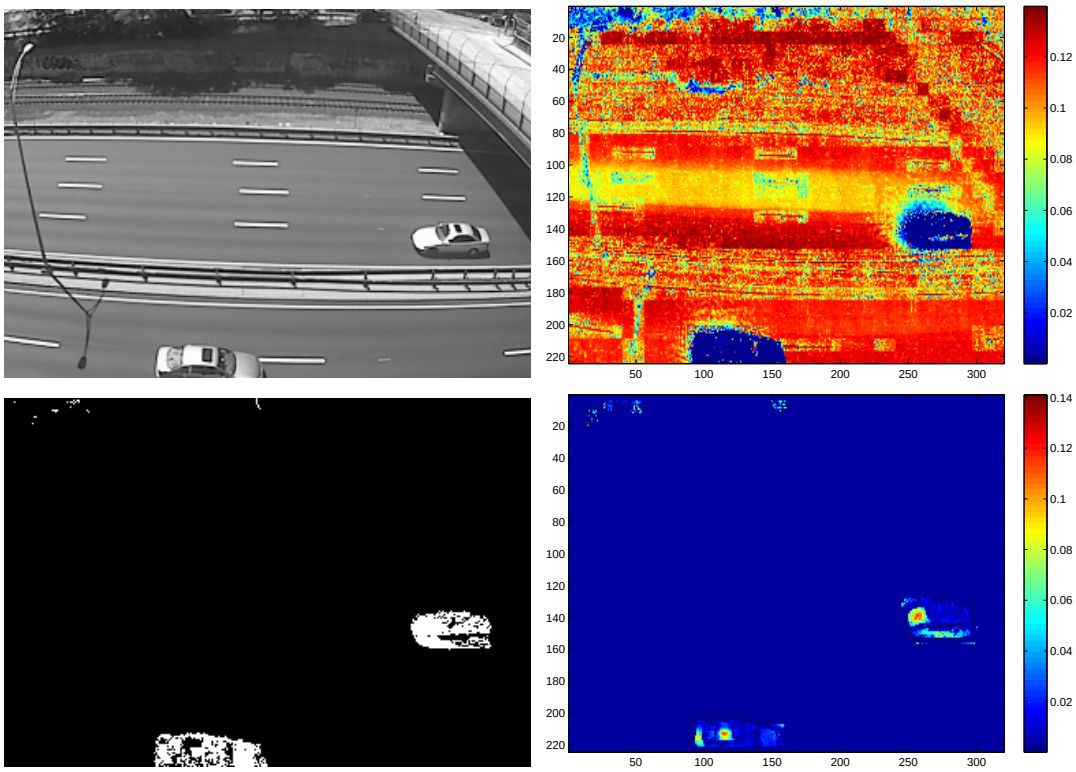

Figure 5.1: Sample frame $I[\boldsymbol{n}]$ (top left) and background probability image $P_{\mathcal{B}}(I[\boldsymbol{n}])$ (top right). The initial detection mask (bottom left) is used to compute the foreground probability image $P_{\mathcal{F}}(I[\boldsymbol{n}])$ (bottom right) which acts as a spatially variable threshold biasing term. Red regions indicate high probability while blue denotes low probability.

iterations and thus the process may be terminated in a reasonable amount of time. This is illustrated in Fig. 5.2. The first detection mask shown was derived by thresholding $P_{\mathcal{B}}$ alone. Once the $P_{\mathcal{F}}$ estimate is incorporated, there are far fewer false negatives, however the subsequent iterations offer very little additional improvement.

Figure 5.2: Several iterations of background subtraction with foreground modeling. Note a significant reduction of false negatives after the first iteration and minimal improvements afterwards.

Care must be taken with this approach since it induces positive feedback into the

detection process. A high number of false positives in e_0 will be greatly detrimental to the process since they will cause $P_{\mathcal{F}}$ to characterize the background and the errors will quickly compound. On the other hand, false negatives will tend to be corrected as long as at least *some* similar neighbors are correctly detected initially. It is therefore important that the initial threshold be set to discourage false detections at the expense of allowing some misses. Looking ahead, we present a method which ensures this in Chapter 6 but for now, the threshold is determined heuristically.

5.1.3 Example results

The intent of incorporating the foreground model is to grow detected regions corresponding to true moving objects. Misses after thresholding $P_{\mathcal{B}}$ alone can be the result of either camouflaging or outliers in the background model (e.g., a previously missed pixel is accidentally added to $\mathcal{M}$). With a decent estimate for $P_{\mathcal{F}}$, we are able to better discriminate in these difficult situations.

Figures 5.3 and 5.4 depict detection results from two testing sequences. In each, the LIS background model was used to construct $P_{\mathcal{B}}$ and frames were registered with the PC preprocessing step. The improvements of this method over fixed thresholding are clearly visible. Some regions that were badly missed initially have been successfully filled in. The man to the left in Fig. 5.3 and the vehicles in the top right and bottom left of Fig. 5.4 most notably. Not unnoticed is the effect of an increased number of false positives. The random speckle noise present in e_0 is, as mentioned, detrimental to this detection strategy. These errors can be suppressed with further processing, however, as we show in the coming sections of this chapter.

5.2 Markov modeling of detection mask

Inspecting the results presented thus far one notices that the detection masks still have a somewhat unnatural look to them, even after the foreground probability refinement step. Most disturbing visually are the false positives, which tend to be randomly spread out

Figure 5.3: A single frame from the testing sequence *sidewalk* (realigned from a shaky camera) and detection results below. The initial detection result after thresholding $P_{\mathcal{B}}$ (LIT model) on the left and the effect of utilizing foreground modeling in the detection process to the right.

across the image. Intuitively, it would seem wise to discourage these types of errors by inspecting neighboring labels. Formally, the dependence on neighboring labels can be expressed through a Markov random field (MRF) model.

Markov random field models have been used in image and video processing in a variety of applications. In (Aach et al., 1993) and (Aach and Kaup, 1995), MRF models are applied to change detection masks of image pairs: e.g., subsequent frames in video. In those treatments, the test statistics were based on pixel-wise image differences. In this thesis, we apply similar methods to background subtraction in a video surveillance scenario where we have arbitrary background and foreground PDFs available, not just observations from two images as those authors had.

The fixed-threshold detection (3.7) considers each pixel independently, i.e., each label is computed regardless of neighboring labels. In other words, the *a priori* probabilities $\pi_{\mathcal{B}}$

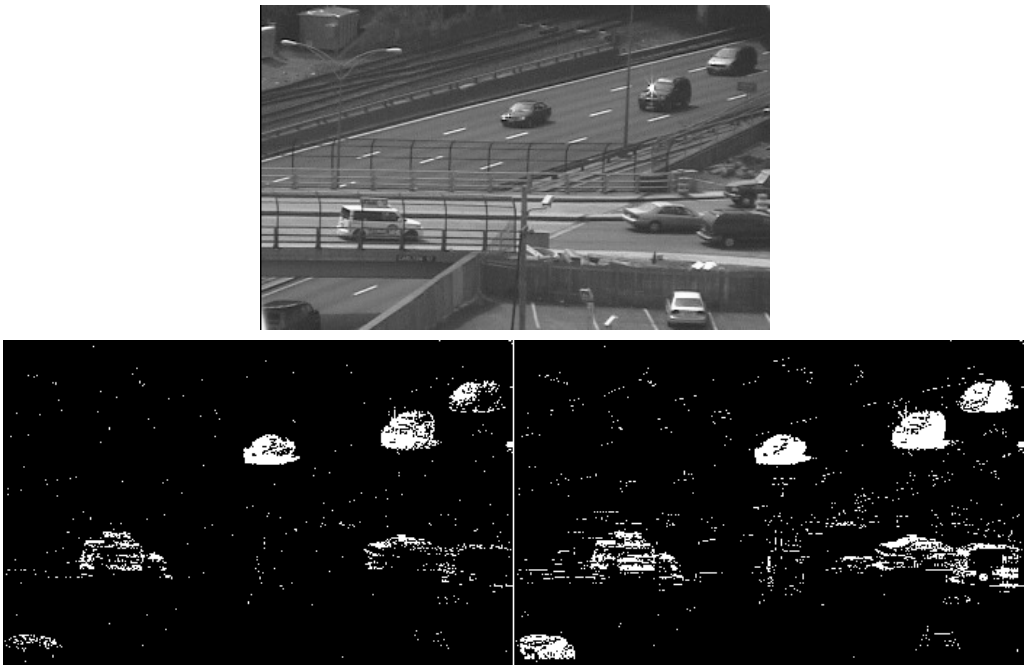

Figure 5.4: A single frame from the testing sequence *highway* and detection results below. The initial detection result after thresholding $P_{\mathcal{B}}$ (LIT model) on the left and the effect of utilizing foreground modeling in the detection process to the right.

and $\pi_{\mathcal{F}}$ are fixed and do not change as the detection process evolves. We propose to model the detection mask by a Markov random field E, with realization e. The main idea behind the new model is the assumption that moving (foreground) objects are usually smooth connected regions. By incorporating local labels in the decision process, we expect to see far fewer scattered false detections as well as smoother moving region boundaries.

5.2.1 Derivation of technique

Here, the background subtraction task is posed as in inverse problem. That is to say, a restriction is imposed on the quantity that we are trying to determine. Specifically, we require the label field e, which is the result of background detection, to be Markovian. By assuming this specific structure *a priori*, we are able to achieve detection results which are much more natural looking than those attainable with previously described methods.

Suppose that the label field realization $e[\boldsymbol{m}]$ is known for all $\boldsymbol{m}$ except $\boldsymbol{n}$. This as-

sumption is reasonable since the estimation process is usually iterative. The background detection task is thus reduced to deciding a label for $e[\boldsymbol{n}]$ only. Let $e^{\mathcal{B}}$ denote the label field produced when $e[\boldsymbol{n}] = 0$ and let $e^{\mathcal{F}}$ denote the case when $e[\boldsymbol{n}] = 1$. Accounting for *a priori* probabilities of the label at $\boldsymbol{n}$, the decision rule for the configuration of e is then

$$\frac{\Pr\left\{\mathcal{I} = I \mid e^{\mathcal{B}}\right\}}{\Pr\left\{\mathcal{I} = I \mid e^{\mathcal{F}}\right\}} \underset{\mathcal{F}}{\overset{\mathcal{B}}{\lessgtr}} \eta \frac{\Pr\left\{E = e^{\mathcal{F}}\right\}}{\Pr\left\{E = e^{\mathcal{B}}\right\}}. \tag{5.3}$$

Note that $\Pr\left\{\mathcal{I} = I | e^{\mathcal{B}}\right\}$ is the probability that the whole random field $\mathcal{I}$ assumes realization I, given label field realization $e^{\mathcal{B}}$, and similarly for for $\Pr\left\{\mathcal{I} = I | e^{\mathcal{F}}\right\}$. The ratio of these joint probabilities can be simplified if we allow the intensities, while dependent on the label field, to be mutually independent spatially. That is,

$$\Pr\left\{\mathcal{I} = I \mid e\right\} = \prod_{\boldsymbol{m}} \Pr\left\{\mathcal{I}[\boldsymbol{m}] = I[\boldsymbol{m}] \mid e[\boldsymbol{m}]\right\}. \tag{5.4}$$

Since $\Pr\left\{\mathcal{I} = I | e^{\mathcal{B}}\right\}$ and $\Pr\left\{\mathcal{I} = I | e^{\mathcal{F}}\right\}$ differ only at $\boldsymbol{n}$, common marginal terms in (5.4) corresponding to $\boldsymbol{m} \neq \boldsymbol{n}$ cancel out and the left-hand side of (5.3) reduces to

$$\frac{\Pr\left\{\mathcal{I} = I[\boldsymbol{n}] \mid e^{\mathcal{B}}\right\}}{\Pr\left\{\mathcal{I} = I[\boldsymbol{n}] \mid e^{\mathcal{F}}\right\}} \equiv \frac{P_{\mathcal{B}}(I[\boldsymbol{n}])}{P_{\mathcal{F}}(I[\boldsymbol{n}])}.$$

Now we focus on the right-hand side of (5.3). Since E is a Markov random field, the *a priori* probabilities are Gibbsian of the following general form (Geman and Geman, 1984):

$$\Pr\{E = e\} = \frac{1}{Z} \exp\left(\frac{-1}{T} \sum_{c \in C} \mathrm{V}(c)\right) \tag{5.5}$$

where Z and T are normalization and natural temperature constants respectively. While Z gets canceled in the prior ratio, T is a parameter to be set by algorithm designer. The potential function, $\mathrm{V}(c)$, operates on cliques, c, in the set of all cliques in the image, C. In this work, we take C to include all 2-element cliques of the second-order Markov neighborhood. Since the potential function operates on 2-element cliques only, we may use

the notation

$$\sum_{c \in C} \mathrm{V}(c) \equiv \sum_{\{\boldsymbol{m},\boldsymbol{n}\} \in C} \mathrm{V}(\boldsymbol{m},\boldsymbol{n}). \tag{5.6}$$

With this clique structure defined, the sites $\boldsymbol{m}$ in (5.6) correspond to the eight immediate neighbors of $\boldsymbol{n}$. Extensions to higher neighborhood orders are straightforward, however, this structure leads to a very simple implementation, as we shall show shortly.

Choosing a suitable potential function, $\mathrm{V}(\boldsymbol{n},\boldsymbol{m})$, is crucial to the model's effectiveness. The potential should be low when the label field e exhibits continuity, resulting in a high probability. Conversely, severe fragmentation of the label field e should incur high values of the potential, driving the probability down. Since the labels are binary, we choose to use the Ising potential function (Geman and Geman, 1984)

$$\mathrm{V}(\boldsymbol{n},\boldsymbol{m}) = \begin{cases} 0 & \text{if } e[\boldsymbol{n}] = e[\boldsymbol{m}] \\ 1 & \text{if } e[\boldsymbol{n}] \neq e[\boldsymbol{m}] \end{cases}$$

In effect, a penalty of 1 is incurred if the label at site $\boldsymbol{m}$ label differs from that at site $\boldsymbol{n}$.

With Z canceled, the ratio of Gibbs priors in (5.3) becomes

$$exp\left\{ \frac{1}{T}\left[\sum_{\{\boldsymbol{n},\boldsymbol{m}\}} \mathrm{V}(\boldsymbol{n},\boldsymbol{m})\right]_{e^{\mathcal{B}}} - \frac{1}{T}\left[\sum_{\{\boldsymbol{n},\boldsymbol{m}\}} \mathrm{V}(\boldsymbol{n},\boldsymbol{m})\right]_{e^{\mathcal{F}}} \right\}.$$

Since $e^{\mathcal{F}}$ and $e^{\mathcal{B}}$ differ only at $\boldsymbol{n}$ by definition, all terms in the above summations are identical except for those including $\boldsymbol{n}$. With the potential function as defined, these two summations count the number of dissimilar neighbors of $\boldsymbol{n}$ for each case. The number of stationary (background) and moving (foreground) neighbors shall be denoted $N_{\mathcal{B}}[\boldsymbol{n}]$ and $N_{\mathcal{F}}[\boldsymbol{n}]$ respectively. The new hypothesis test becomes:

$$\boxed{\frac{P_{\mathcal{B}}(I[\boldsymbol{n}])}{P_{\mathcal{F}}(I[\boldsymbol{n}])} \underset{\mathcal{F}}{\overset{\mathcal{B}}{\gtrless}} \eta \cdot \exp\left(\frac{1}{T}\left(N_{\mathcal{F}}[\boldsymbol{n}] - N_{\mathcal{B}}[\boldsymbol{n}]\right)\right)} \tag{5.7}$$

The effect of incorporating the new prior information into the likelihood ratio test is readily apparent. Detected moving neighbors bias the threshold toward declaring $\mathcal{F}$

whereas stationary neighbors bias the threshold toward declaring $\mathcal{B}$. Again, this procedure may be viewed as a spatially variable thresholding strategy, as illustrated in Fig. 5.5. The constant $1/T$ can be thought of as a threshold variance (termed loosely), which in our experiments is set to about 1.5.

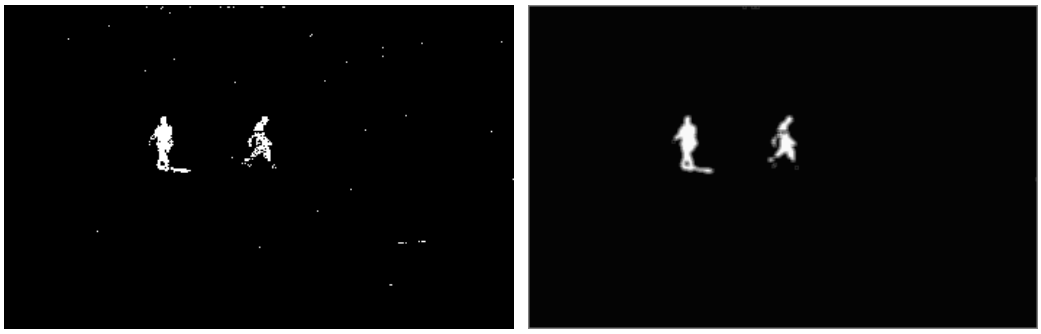

Figure 5.5: Sample frame $I[\boldsymbol{n}]$ (left), initial detection result $e_0[\boldsymbol{n}]$) (center), and log of MRF prior (right). Light regions indicate high probability and a bias towards declaring $\mathcal{F}$ and dark regions denote the converse.

5.2.2 Notes on implementation

The detection process of (5.7) takes an existing label field as input and the result is optimized iteratively according to the Markov prior weighting function. The initial field represents our initial "best guess" of the true result. In this work, we have used a deterministic relaxation method to optimize the detection result known as *iterated conditional modes* (ICM) (Besag, 1986). An alternative approach would be to use a stochastic relaxation technique, such as simulated annealing, which will converge to the true result with high likelihood (Geman and Geman, 1984). This is the approach taken in (Migdal and Grimson, 2005) where the detection is formulated as a *maximum a posteriori* (MAP) estimation problem. Stochastic relaxation techniques are generally slow to converge, however. We opted to use ICM because it is fast, simple, and it provides very good detection results. We have noticed that the MRF detection result tends to converge quickly - usually in less than ten iterations.

The optimization consists of counting $N_{\mathcal{F}}[\boldsymbol{n}]$ and $N_{\mathcal{B}}[\boldsymbol{n}]$ according to the current la-

bel field and calculating the prior weight, $\exp\left(\frac{1}{T}\left(N_{\mathcal{F}}[\boldsymbol{n}] - N_{\mathcal{B}}[\boldsymbol{n}]\right)\right)$. With the quantities $P_{\mathcal{B}}(I[\boldsymbol{n}])$ and $P_{\mathcal{F}}(I[\boldsymbol{n}])$ already stored from previous steps, the test described by (5.7) is carried out and the new label is assigned to $e[\boldsymbol{n}]$. The test then moves to the next pixel and repeats. The label field is updated "in-place", which increases the speed of convergence. The scanning paths alternate to prevent a biased propagation of the label field in one of the scanning directions.

5.2.3 Detection improvement with MRF

The new model penalizes local dissimilarities in the detection mask which are characteristic of random scattered false positives and jagged object boundaries. When we apply (5.7) to the background subtraction results of Chapter 4 (still assuming that $P_{\mathcal{F}}$ is constant) we notice a substantial visual improvement. Figures 5.6 and 5.7 illustrate this.

Figure 5.6: A single frame from the testing sequence *sidewalk* (realigned from a shaky camera) and detection results below. The initial detection result after thresholding $P_{\mathcal{B}}$ (left) and the effect of incorporating Markov prior (right).

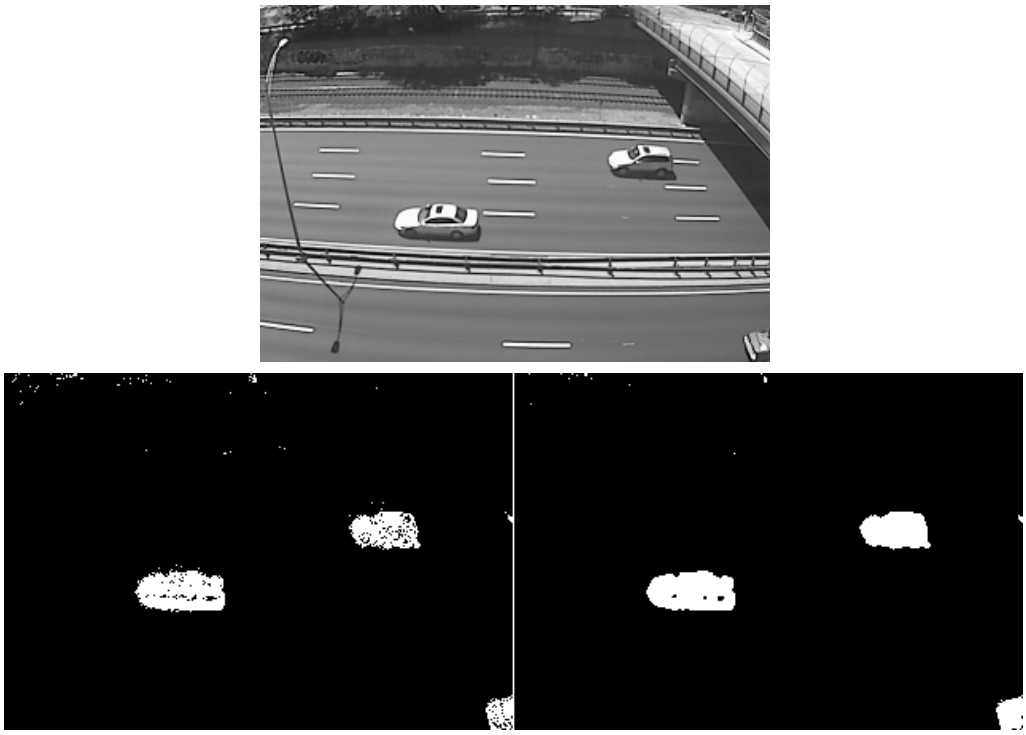

Figure 5.7: A single frame from the testing sequence *pike_1* and detection results below. The initial detection result after thresholding $P_{\mathcal{B}}$ (left) and the effect of incorporating Markov prior (right).

5.3 Incorporating both methods

As the results in the previous sections have demonstrated, probabilistic modeling of the parameters $\pi_{\mathcal{B}}$, $\pi_{\mathcal{F}}$, and $P_{\mathcal{F}}$ can offer an improvement in the background subtraction task. We can discourage misses by incorporating an explicit foreground object model, and false positives can be suppressed by modeling the detection mask as an MRF. It would be desirable to use both models to our advantage. Initially, however, there is a slight ambiguity as to how to incorporate both methods simultaneously.

As (5.7) might indicate, both the foreground probability and the prior weight could both be determined and then tested simultaneously and the process iterated. The difficulty we have noticed with this approach however is that, like the foreground modeling stage itself, the result does not tend to converge quickly. Recall that in Section 5.1.2 it is suggested

that the foreground model refinement process be stopped after only a few iterations. When the two probabilities are estimated and tested in the same step, the MRF stage does not work as designed. The label field should be allowed to converge with the MRF process before a new probability (likelihood ratio) is tested.

Consequentially, we are inclined to separate the methods and perform them serially rather than in parallel. Again from the discussion of Section 5.1.2, recall that the estimate of $P_{\mathcal{F}}$ will erroneously characterize the background when there are false detections present and that this should be avoided. We have seen that the MRF stage will suppress many false positives and one may be inclined to perform this stage first. Alternatively, the operations may be performed in the reverse order; first the LR is refined, then the resulting detection mask is 'smoothed' by the MRF prior to obtain the final result.

Through experimentation and comparison of the three methods described - jointly updated, MRF followed by foreground model, and foreground model followed by MRF - it was observed that the latter approach consistently performed better than the others. This should be intuitively satisfying. Since the MRF process is formulated specifically as an inverse problem, as a constraint on the final result, it makes sense that it should be invoked last.

In the following sets of images, the effects of incorporating both detection modalities are presented. In Figures 5.8 an 5.9, a sample frame from an image sequence is shown alongside the initial detection result, the result of the second stage (foreground model), and the final result (after MRF step). Notice that although the *number* of false detections is increased after the second stage, the *nature* of them is the same. That is, the false detections still tend to be randomly scattered as opposed to being tightly clustered. Thankfully, since the imposition of Markovianity upon the detection mask suppresses this type of behavior, the final result is free of false detections for the most part.

Figure 5.8: Background detection result on *highway* sequence. Thresholding of $P_{\mathcal{B}}$ (LIT model) alone (top right), inclusion of $P_{\mathcal{F}}$ (bottom left), and final MRF step (bottom right).

Figure 5.9: Background detection result on *sidewalk* sequence. Thresholding of $P_{\mathcal{B}}$ (LIT model) alone (top right), inclusion of $P_{\mathcal{F}}$ (bottom left), and final MRF step (bottom right).

5.4 Performance evaluation on synthetic sequences

In order to objectively evaluate the methods proposed, receiver operating characteristic (ROC) profiles were determined empirically. The ROC curve displays the decision rule's performance in terms of the error rates FPR and FNR (Section 3.2). With a very low threshold, there tend to be few false alarms but many misses; the converse is true for a very high threshold. Generally, neither scenario is desirable. The ROC curve characterizes the behavior of the classifier as the threshold parameter is swept between the two extremes.

In order to quantify the false alarm and miss rates of each detection method, one must know the ground truth. It is possible to segment real sequences by hand however, it is unrealistic to do so for more than a few frames. In the evaluation we have conducted, the reported error rates are average rates observed over 100 frames. A more practical way to obtain ground truth data for such a large number of frames is to generate synthetic image sequences.

The aim of this evaluation is to characterize the newly posed detection methods irrespective of the scene or background model. To this end, three synthetic sequences were created. Each was designed for use with one of the background models described in Chapter 4. The three testing sequences will be briefly described forthwith, and this chapter will conclude with the experimental results.

5.4.1 Synthetic sequences

Static background - $\mathcal{M}_0$

The first synthetic sequence, dubbed *synth_M0*, was designed for use with the simplified background model $\mathcal{M}_0$. A sample frame and its corresponding ground truth detection mask are provided in Fig. 5.10. A single frame from a real sequence was repeated 100 times with noise to emulate a truly static background captured by one of our surveillance cameras. The additive Gaussian noise is white (independent and identically distributed) with a mean $\mu_n = 0$ and variance $\sigma_n^2 = 6$. After the addition of noise, the image was quantized to the range $[0, 255]$. Superimposed upon this background are five textured blobs, the locations

of which were randomly generated. By randomly disbursing the blobs across the textured background, varying degrees of camouflaging and overlapping were achieved.

Figure 5.10: Sample frame from synthetic sequence *synth_M0* and ground truth label field.

Dynamic background - $\mathcal{M}_1$

The next sequence, *synth_M1*, was built with the same testing data used in (Elgammal et al., 2000). A real video, barren of foreground objects, was used as the background for this sequence. Trees and bushes in the background exhibit texture and move slightly with wind. Three foreground objects, a pedestrian, a truck, and a traffic barrel, were superimposed on the last 100 frames. The first 100 frames were left barren in order to train the model $\mathcal{M}_1$. A sample frame from the synthetic sequence and its corresponding true detection mask are provided in Fig. 5.11.

Shaky background - $\mathcal{M}_2$

The third and final testing sequence, *synth_M2*, was designed for use with the LIS background model. Similar to the previous scenario, this synthetic sequence consists of 100 frames of a real video sequence barren of foreground objects with a truck superimposed. The real sequence was captured with an outdoor mounted camera with a high level of zoom, resulting in a substantial jitter between frames. To estimate the background image B, which $\mathcal{M}_2$ relies on, the following approach was taken. Using the phase correlation method described in Appendix A, the relative displacement among frames was determined

Figure 5.11: Sample frame from synthetic sequence *synth_M1* and ground truth label field.

Figure 5.12: Sample frame from synthetic sequence *synth_M2* and ground truth label field.

to within single pixel accuracy. We identified a subset of the 100 frames with the median horizontal and vertical displacement, i.e., the most centered frames in the sequence. A 1-D temporal median was applied to the frames in this subset to arrive at the estimate for B.

5.4.2 Results and discussion

As previously mentioned, the ROC curve summarizes each method's detection performance in terms of FPR and FNR as the threshold parameter is varied. We have compared the three spatially adaptive thresholding strategies - foreground object modeling, MRF detection mask modeling, and foreground followed by MRF - against non-adaptive, fixed thresholding.

To illustrate the insensitivity of these methods to the particular scene or background PDF, three separate experiments were performed using the three background models presented in Chapter 4: the simple background frame model, $\mathcal{M}_0$, is applied to *synth_M0*; the LIT model, $\mathcal{M}_1$, is applied to *synth_M1*; and the LIS model, $\mathcal{M}_2$,is applied to *synth_M2*. For the models that require a background image to perform detection, $\mathcal{M}_0$ and $\mathcal{M}_2$, a single image is provided and no updating takes place over the 100 frames. This of course means that in a real setting, any errors in estimating B would worsen the performance. For background model $\mathcal{M}_1$, which inherently adapts in time, a selective updating scheme was used. That is, pixels detected as moving are not added to the background model.

The error rates for each of the four detection methods and for each of the three synthetic sequences are presented in Figures 5.13, 5.14, and 5.15. What we notice from these plots is that, regardless of the scene and the background model used, the adaptive detection methods presented in this thesis will outperform fixed thresholding (blue). Over a range of thresholds, the ROC curves corresponding to the adaptive methods are closer to the FPR and FNR axes. The combined foreground-Markov method (cyan) performs the best overall, with Markov alone (red) in a close second place.

The total error rate (TER) plots verify what we might expect. That is, for higher values of threshold, the foreground biased detector (green) will have higher error rates than fixed thresholding since false positives will compound. The cyan curves however reinforce the fact that these compounded errors will tend to be randomly scattered, and they can be aptly suppressed by the MRF stage.

A sample frame from each sequence, along with the ground truth and four detection results, are provided in Figures 5.16, 5.17, and 5.18.

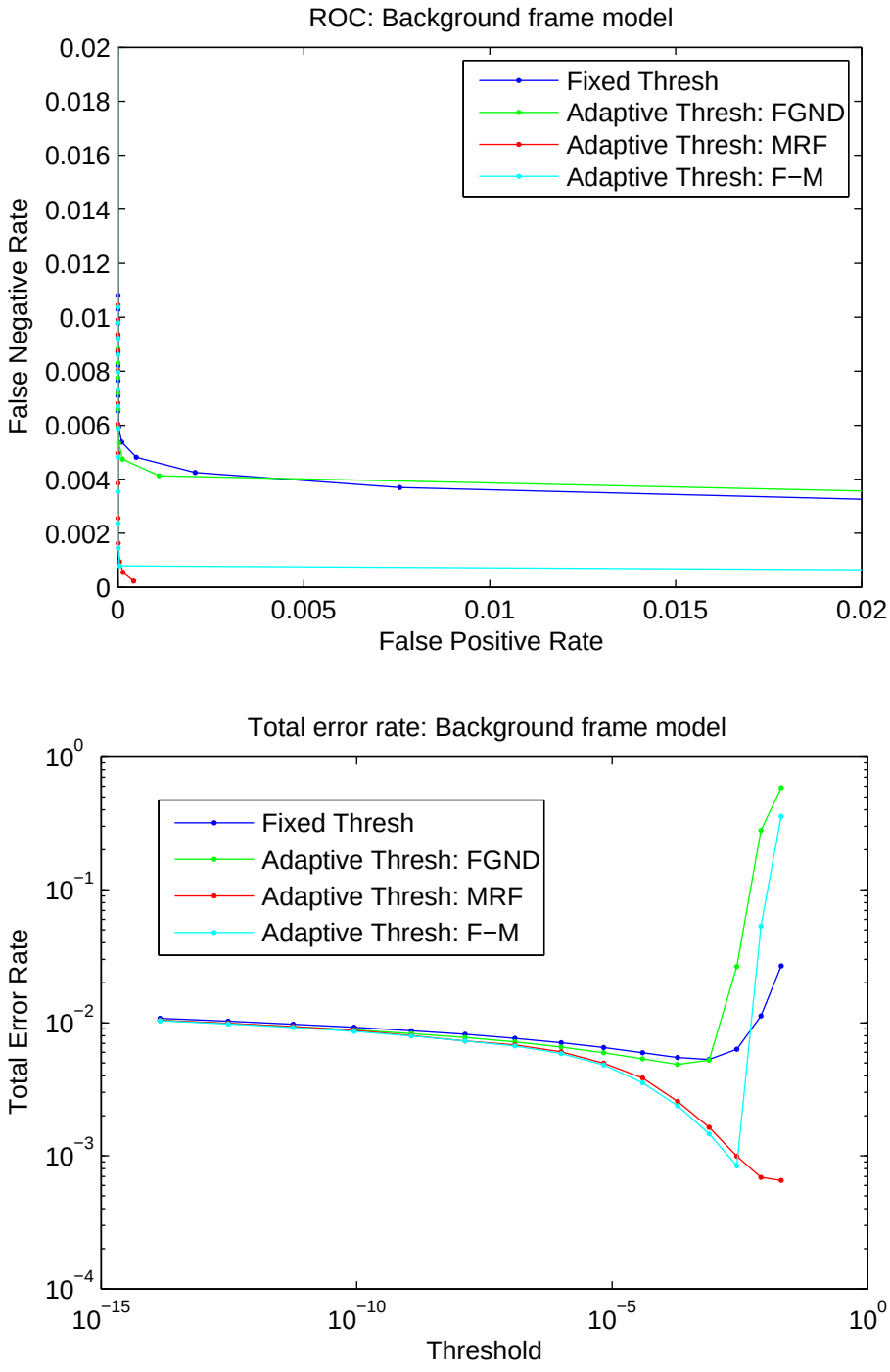

Figure 5.13: Empirical ROC and TER curves for sequence *synth_M0* with background model $\mathcal{M}_0$.

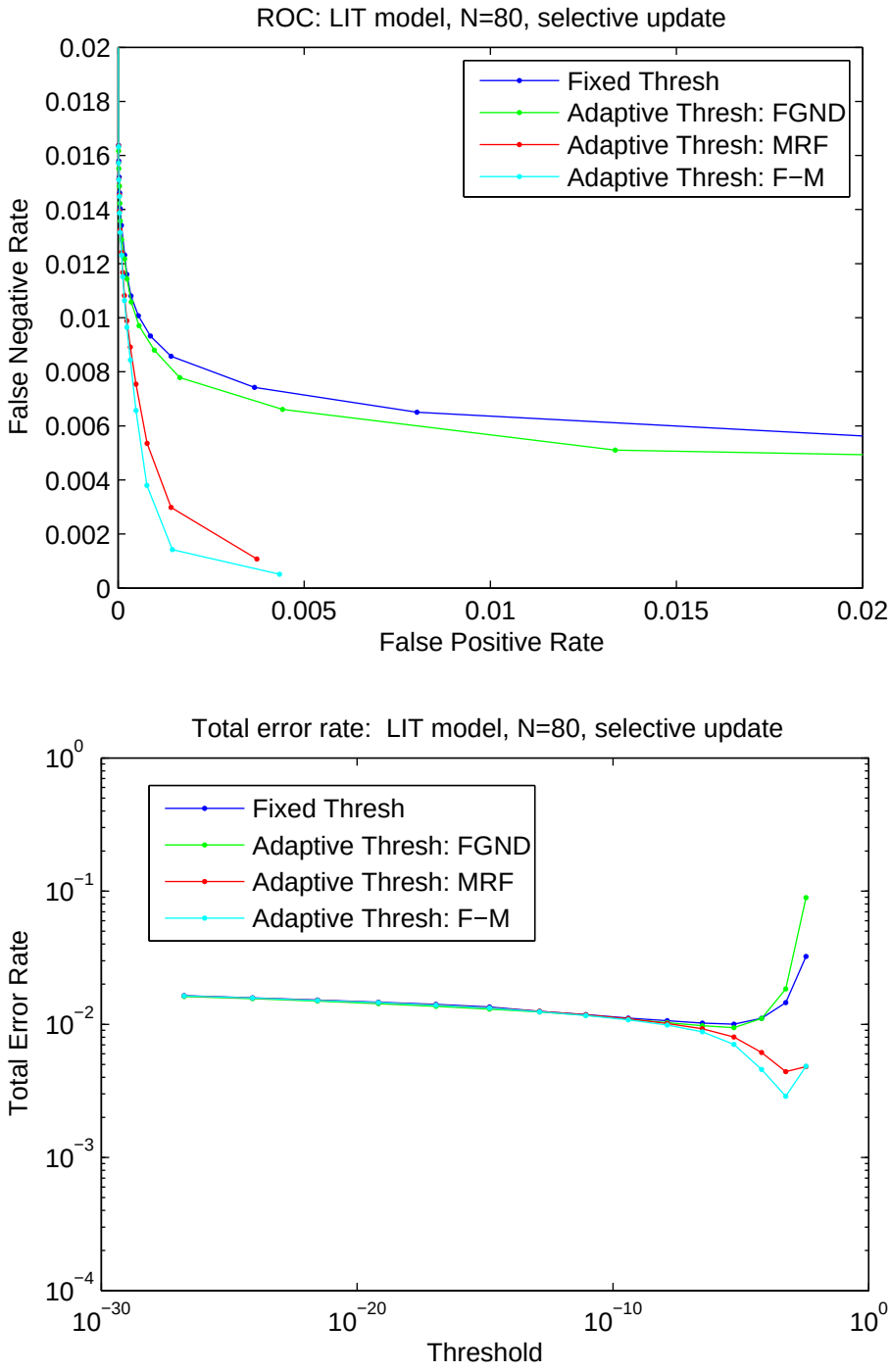

Figure 5.14: Empirical ROC and TER curves for sequence *synth_M1* with background model $\mathcal{M}_1$.

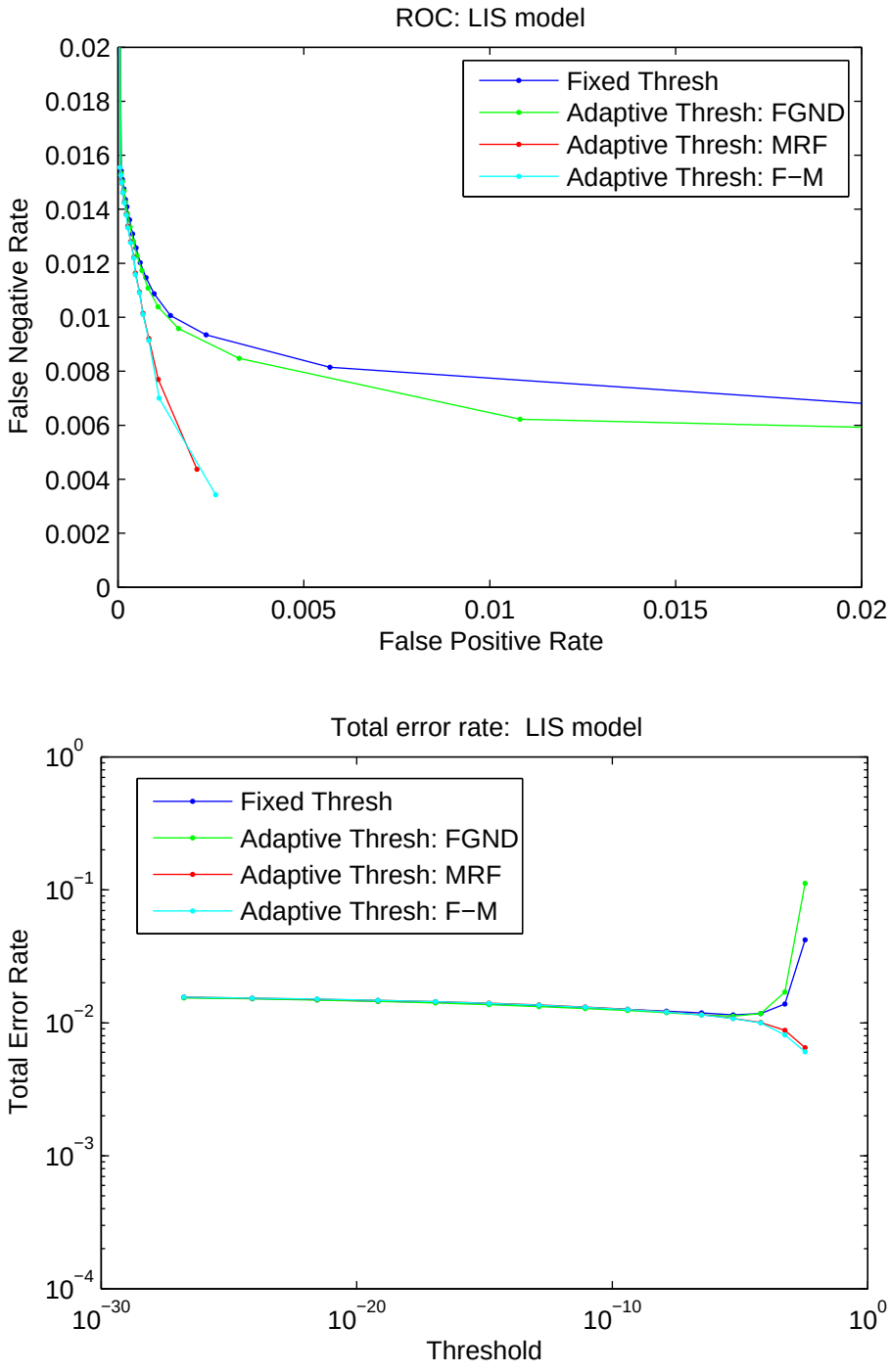

Figure 5.15: Empirical ROC and TER curves for sequence *synth_M2* with background model $\mathcal{M}_2$.

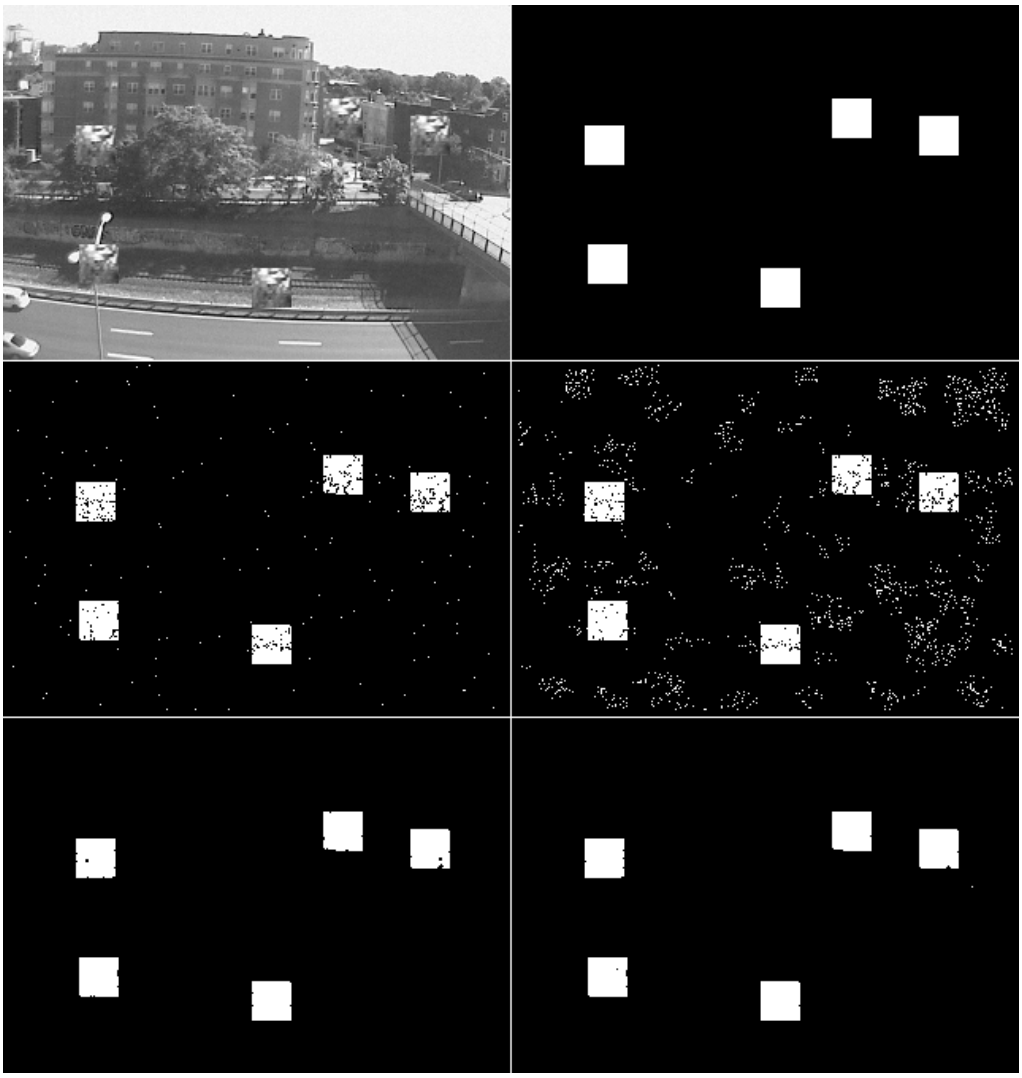

Figure 5.16: Sample detections for sequence *synth_M0* with background model $\mathcal{M}_0$. Top row: frame and ground truth; middle row: fixed threshold and foreground-biased detections; bottom row: MRF prior applied to fixed and foreground-biased detections

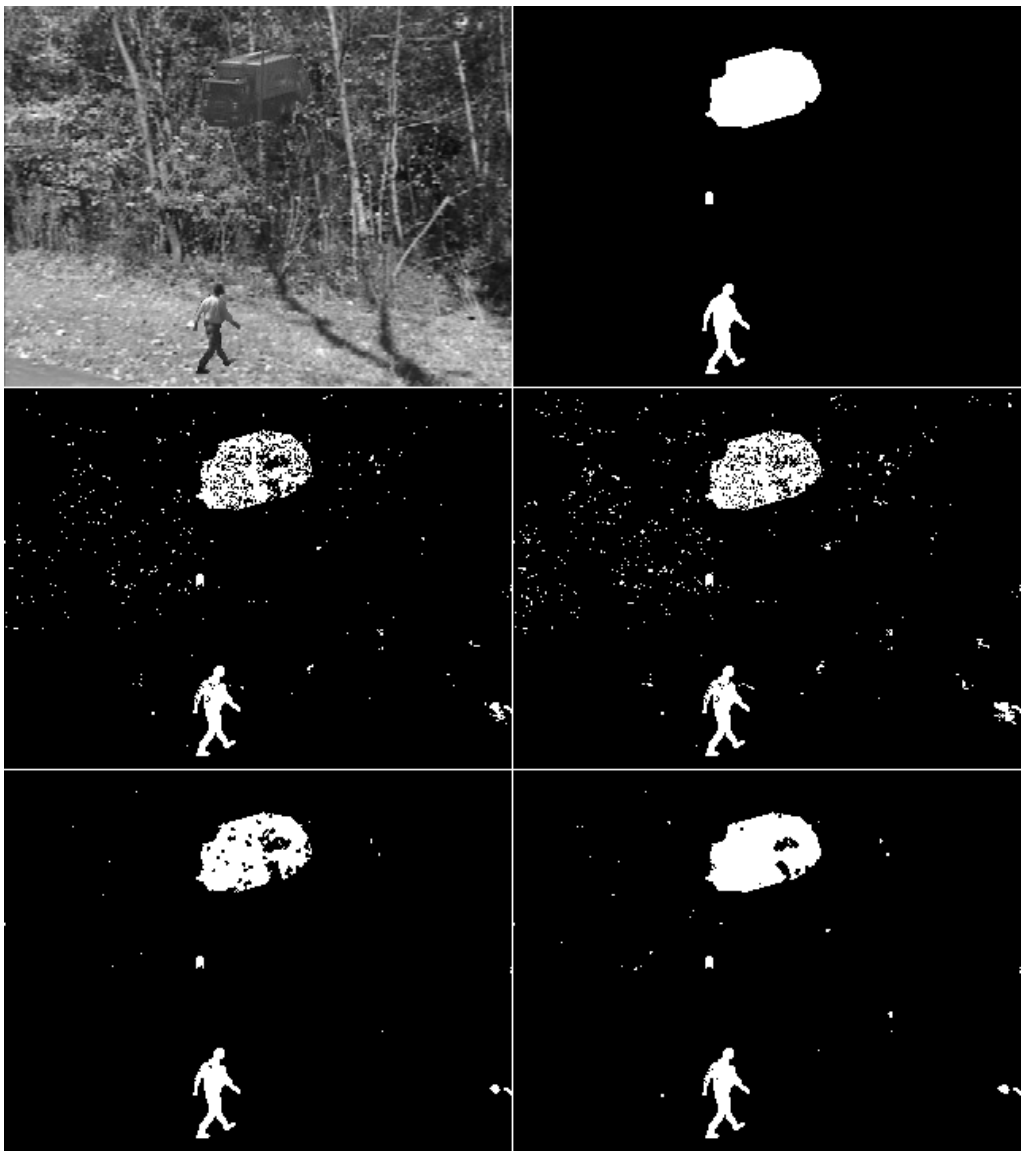

Figure 5.17: Sample detections for sequence *synth_M1* with background model $\mathcal{M}_1$. Top row: frame and ground truth; middle row: fixed threshold and foreground-biased detections; bottom row: MRF prior applied to fixed and foreground-biased detections

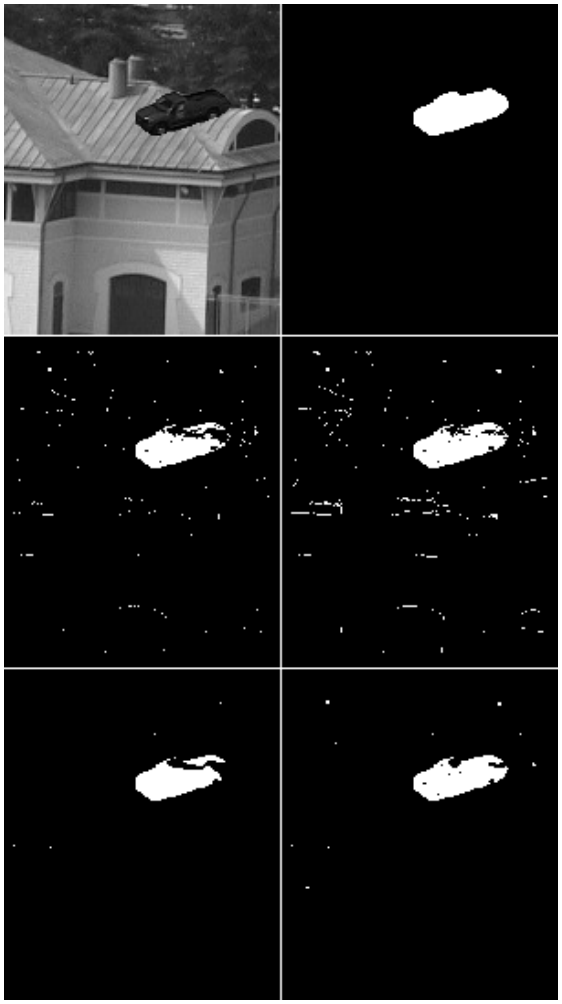

Figure 5.18: Sample detections for sequence *synth_M2* with background model $\mathcal{M}_2$. Top row: frame and ground truth; middle row: fixed threshold and foreground-biased detections; bottom row: MRF prior applied to fixed and foreground-biased detections

Chapter 6

Temporally adaptive detection via FDR control

In terms of detection, we have thus far presented two thresholding procedures which are adaptive in space. Now, we shall present a complementary detection method which adapts in time. The method is a multiple comparisons procedure (MCP) that controls the so-called *false discovery rate* (FDR) (Benjamini and Hochberg, 1995). The false discovery rate is defined as the expected proportion of false alarms to total positive declarations. Recall from Section 3.2 the definition in terms of the random variables V and R is

$$FDR = \mathbb{E}\left\{\mathsf{V}/\mathsf{R}\right\}.$$

Preceding the presentation of the technique in Section 6.2, it is necessary to familiarize the reader with the concept of statistical significance which is pertinent to the method. This chapter concludes with experimental results on synthetic and real video sequences as well as numerical and qualitative evaluations thereof.

6.1 Statistical significance

In MCP formulations, it is common to perform what is known as *significance testing*. In short, by assigning a significance score to each observation, which shall be henceforward designated as its p-score, the observations can be classified as either significant or insignificant.

6.1.1 Definition

The common definition of a p value from the Neyman-Pearson statistical testing viewpoint is *the probability that a less likely outcome than the current observation could occur given*

a null hypothesis (Lehmann, 1986). In our treatment, this general definition is followed faithfully. Interpreted mathematically, this translates to

$$p = f(x) = \int_{\mathcal{R}_1(x)} P_0(y)dy$$
$$\mathcal{R}_1(x) = \{y \,|\, P_0(y) \leq P_0(x)\}. \tag{6.1}$$

This definition is illustrated in Fig. 6.1.

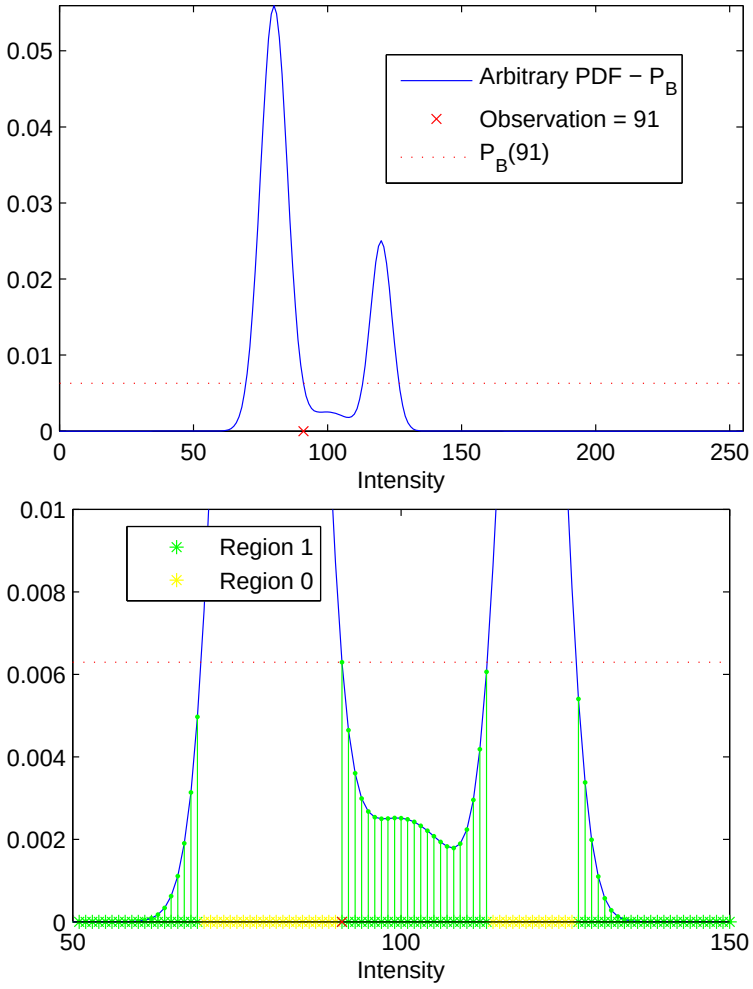

Figure 6.1: Definition of a significance score. The highlighted (green) area corresponds to the probability of a less likely occurrence than the current observation (91).

6.1.2 Properties

Being a deterministic function of the random variable X, the significance score is itself a random variable; we shall denote the random quantity as $\mathcal{P}$ with p being it's realization. From the definition (6.1) it can be readily seen that the significance score p is in fact a probability, and therefore it must exist on the interval $[0, 1]$. Notice that the state space is not intensity as it is for x (or $I[\boldsymbol{n}]$). A crucial property of a significance score is that it is uniformly distributed under the null hypothesis. That is, when $X \sim P_0(x)$ it must follow that $\mathcal{P} \sim \text{Unif}[0, 1]$. This proof of this property is given below in (6.2).

$$\begin{aligned}
\text{CDF}(p) &= \Pr\{\mathcal{P} \leq p\} && \text{(6.2a)}\\
&= \Pr\{f(X) \leq f(x)\} && \text{(6.2b)}\\
&= \Pr\{P_0(X) \leq P_0(x)\} && \text{(6.2c)}\\
&= \Pr\{X \in \mathcal{R}_1(x)\} && \text{(6.2d)}\\
&= \int_{\mathcal{R}_1(x)} P_0(y)dy && \text{(6.2e)}\\
&= f(x) = p \quad \blacksquare && \text{(6.2f)}
\end{aligned}$$

Lines (6.2a) and (6.2b) are by the definition of the cumulative density function (CDF) and of the significance score; (6.2c) follows from the fact that if $P_0(x)$ decreases, $f(x)$ must decrease; (6.2d) follows directly from the definition of $\mathcal{R}_1$; and (6.2e) holds true if X is in fact distributed on P_0. Since the CDF, $\Pr\{\mathcal{P} \leq p\} = p$, is linearly increasing over the range $[0, 1]$, the PDF is constant i.e., it is uniform.

There is one other property of the significance score which is important but not necessary for FPR or FDR controlling procedures. That property is that under H_1, the significance scores be clustered around zero. Since it is not a strict requirement this property does not warrant a rigorous proof, but it does deserve some consideration. It is easy to see that when the true underlying PDFs are well separated, observations that are highly probable according to P_1 will have low p-scores. However, when P_0 and P_1 overlap considerably,

observations which truly arise from source H_1 may elicit p-scores which are not clustered around zero. In terms of background subtraction, this may occur when a foreground object closely resembles the background - when camouflaging is present.

6.1.3 Relation to PDF thresholding

Now, let us apply the idea of significance testing to the background subtraction methods we have already presented in this thesis. Notice that the definition of the significance score depends only on the PDF of the null hypothesis, P_0. This is analogous to the simplified LRT whereby the background PDF is compared to a threshold: $P_{\mathcal{B}} \gtrless \theta$. The significance score as defined in (6.1) has a profound meaning in this context. It is exactly the false alarm probability when the threshold θ is set equal to $P_0(x)$. When $\theta = P_0(x)$, it is clear that $\mathcal{R}_1$ denotes the region where H_1 is declared - its complement, the region where H_0 is declared. The integral (6.1) therefore corresponds to

$$\Pr\left\{\text{declare } H_1 \,\middle|\, H_0 \text{ true}\right\} = FPR.$$

Herein lies the power of statistical significance testing. The error rate (FPR) and significance scores are intimately related. To control the global FPR to be within some level α, declare all observations H_1 for which $p(x) \leq \alpha$.

In this thesis, the antecedent framework has been binary hypothesis testing at each pixel. According to (3.7), a global threshold θ is applied to every test, regardless of the probability distribution $P_{\mathcal{B}}$. In general, each null hypothesis will have a different PDF. This implies that thresholding the PDFs with a global, θ, will not have any direct effect on the overall error probability. In MCP terms this is known as *uncorrected* testing (Lehmann, 1986). That is, the error probability is not controlled for each individual test. By testing in the *significance* domain rather than in the traditional probability domain, tighter control of the error rate is possible.

6.1.4 Special case: $\mathcal{M}_0$

As we have just mentioned, the transformation between the probability and significance domains is space variant. This is because the background PDF will have a different shape at each pixel. For an arbitrary PDF such as a non-parametric density there is no compact functional form for $p = f(x)$. Absent a closed form expression, we must revert to calculating the integral of (6.1) explicitly. This requires that we compute *all* 256 values of P_0 and sum those that fall below the current observation.

The case of $\mathcal{M}_0$ is special, however, in that although $P_{\mathcal{B}}$ is different at each pixel, it has exactly the same shape everywhere. Recall that for $\mathcal{M}_0$, $P_{\mathcal{B}}$ is distributed normally with variance σ^2 and mean $B[\boldsymbol{n}]$. For now, allow the argument x to represent the absolute difference $|I[\boldsymbol{n}] - B[\boldsymbol{n}]|$, i.e., the mean is subtracted from each observation. In the case of the zero mean Gaussian, the function that defines the significance score, $f(x)$, translates to the area under both tails of the distribution. In terms of the zero mean Gaussian CDF this takes the form

$$\begin{aligned} p = f(x) &= 2\left(1 - \mathrm{CDF}(x)\right) \\ &= \mathrm{erfc}\left(x/\sqrt{2\sigma^2}\right) \end{aligned} \tag{6.3}$$

where erfc denotes the *complementary error function*. In practice, the values of $f(x)$ can be precomputed according to (6.3) for positive integral values of x and looked up in a list in the same manner that the PDF kernel function values were in previous chapters.

For the simple background model, $\mathcal{M}_0$, there is a space-invariant relationship between a global threshold, θ, and a global significance, p (FPR). Explicitly, the relationship is

$$\begin{aligned} \theta &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-(\mathrm{erfc}^{-1}\, p)^2\right) \\ p &= \mathrm{erfc}\left(\sqrt{\tfrac{-1}{2}\ln\left(2\pi\sigma^2\theta^2\right)}\right) \end{aligned} \tag{6.4}$$

where erfc^{-1} denotes the *inverse complementary error function*.

6.2 Controlling the false discovery rate

As we have seen in Section 6.1.3, the significance domain is convenient since controlling the global error rate is straightforward. Recall that controlling the global false positive rate to some level, i.e.,

$$FPR = \mathbb{E}\{\mathsf{V}\}/M \leq \alpha$$

generally requires controlling the individual false alarm thresholds at each trial such that

$$\Pr\left\{\text{declare } H_1 \;\middle|\; H_0\right\} \leq \alpha.$$

Such types of decision rules can lead to a large number of false detections. The FPR cannot be made arbitrarily small, however, since doing so will increase the false negative rate.

The concept behind the FDR controlling procedure is to control the expected proportion of positive detections that are falsely declared, i.e.,

$$FDR = \mathbb{E}\{\mathsf{V}/\mathsf{R}\} \leq \alpha$$

It is clear that any method which controls the FDR must also control the FPR since $\mathsf{R} \leq M$, thus $\mathbb{E}\{\mathsf{V}\}/M \leq \mathbb{E}\{\mathsf{V}/\mathsf{R}\}$. The procedure for controlling the FDR is as follows (Benjamini and Hochberg, 1995).

1. Assign a p value to each observation.
2. Sort all p values in ascending order, $p_1, p_2, \ldots, p_M$
3. Find the largest index, i, such that $p_i \leq \frac{i}{M}\alpha$. Call the value corresponding to this index p^*.
4. Declare observations p_i significant for all $p_i \leq p^*$.

For the proof of this method, we refer the reader to the original article (Benjamini and Hochberg, 1995).

The power and the adaptive nature of the procedure can be seen in Fig. 6.2. The ordered p-scores are plotted in blue and the FDR and FPR thresholds are shown as dashed

lines. The flat section of the graph, where the p-scores are low, corresponds primarily to

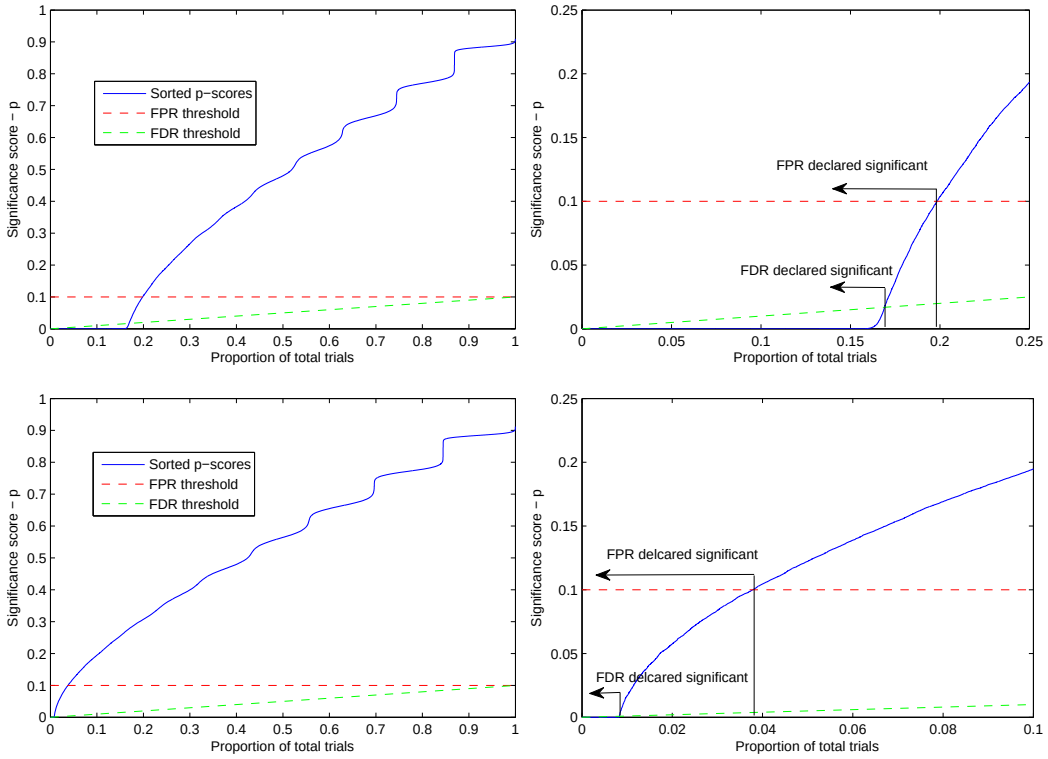

Figure 6.2: Illustration of FDR procedure with varying $H1$ density - approximately 16% (top) and approximately 1% (bottom). Detail in the low significance region is provided in the graphs to the right. Here the parameter α is set to 10% for illustrative purposes.

the low significance scores produced by true H_1's. The section that is linearly increasing (approximately) from 0 to 1 corresponds primarily to the p-scores of true H_0's.

Clearly the FDR threshold will always declare fewer H_1's than the FPR threshold. This follows intuitively from the definition of the false discovery rate: since $\mathsf{R} \leq M$,

$$\mathbb{E}\{\mathsf{V}\}/M \leq \mathbb{E}\{\mathsf{V}/\mathsf{R}\}.$$

Thus, any method which controls the FDR to within α also necessarily controls the FPR to be be less than α. Additionally, we can see that the FDR procedure will adapt to the true density of H_1's. Again this is inherent because of the dependence on the quantity $\mathsf{R} = \mathsf{V} + \mathsf{S}$, which clearly incorporates the proportion of true H_1 pixels (S).

6.3 Results

The procedure described above maps nicely to the task of background subtraction. With $P_{\mathcal{B}}$ defined at every pixel (equivalent to P_0 in Section 6.1) we wish to test each of our current observations, i.e., the intensity values present in the current frame I. Significance scores are determined for each trial (pixel) and the p-scores are sorted in a list. The significance threshold, p^*, is then determined using the aforementioned procedure. Observations whose p-score lie below the threshold are declared $\mathcal{F}$ (significant) and those above are declared $\mathcal{B}$ (insignificant).

In this section the FDR controlling procedure is applied to synthetic and real image sequences. From the results presented, one can clearly see the adaptivity and detection power afforded by this technique.

6.3.1 Synthetic density experiment

For this experiment, we have created synthetic sequences similar to *synth_M0* as described in Section 5.4.1. Three such sequences were synthesized, each with the same static background and with 100 frames containing randomly disbursed foreground objects. These three new testing sequences vary from one another in the number of superimposed foreground objects - i.e., the true density of H_1. Sample frames from each of the three sequences are presented in Fig. 6.3. By varying the density of true foreground objects, we illustrate the advantages of the FDR controlling procedure over the non-adaptive FPR controlling procedure.

Table 6.1: Error rates for controlled FPR and controlled FDR on synthetic sequence with background model $\mathcal{M}_0$.

		controlled FPR		*controlled FDR*		
# objects	α	FPR	FNR	median(p^*)	FPR	FNR
1	0.01	$7.8 \cdot 10^{-3}$	$6.8 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$	$9.4 \cdot 10^{-5}$	$9.8 \cdot 10^{-4}$
10	0.01	$7.3 \cdot 10^{-3}$	$7.0 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	$4.7 \cdot 10^{-4}$	$9.1 \cdot 10^{-3}$
25	0.01	$6.4 \cdot 10^{-3}$	$1.6 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$	$1.8 \cdot 10^{-3}$	$1.8 \cdot 10^{-2}$

Table 6.1 summarizes the results of the experiment. In all cases, the parameter α is

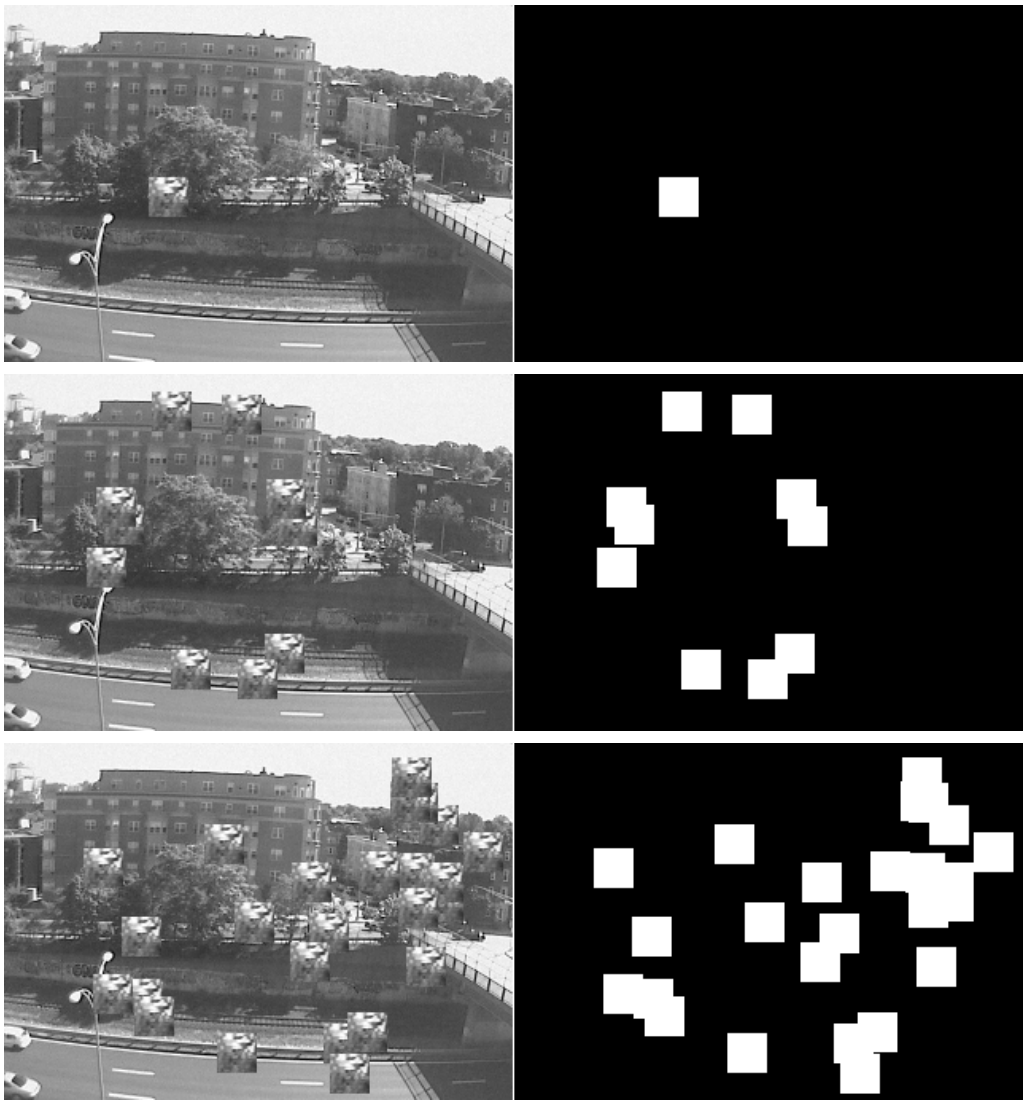

Figure 6.3: Sample frames and ground truth label fields from three synthetic sequences for FDR experiment - 1 object (top), 10 objects (middle), 25 objects (bottom).

held constant at 1%; α corresponds to the false alarm rate for the fixed threshold test and to the false discovery rate for the adaptive threshold test. The significance threshold is exactly α in the fixed threshold case and p^* for the adaptive case. The value p^* can vary from frame to frame. However, since the object density is relatively constant in each experiment, the variation is only occasional. Therefore, the median value over the 100

frames is reported for comparison. Notice that the FDR controlling procedure allows the significance threshold to vary with object density.

The gain in detection power can be seen both in the reported error rates and in the sample detection results presented in Fig. 6.4. While the false negative rates stay on the same order for both FPR and FDR as the object density varies, the false alarm rate is always lower for the FDR procedure - by as much as two orders of magnitude.

6.3.2 Application to real sequences

We have seen that with the FDR method, we can set a non-arbitrary parameter to achieve a strictly bounded error rate. In the previous chapters, there was no clear "best" way to select the detection threshold, θ. Recall too that the foreground biased detection strategy presented in Section 5.1 requires a low false positive rate to begin with, otherwise non-foreground regions may be erroneously grown. With this new technique at our disposal, we can find an initial detection mask with a bounded false positive rate. Moreover, by bounding the *false discovery rate* we are ensured an even lower FPR when the scene allows (when the object density is low) without any substantial penalty in terms of misses.

Having shown the detection power of the FDR procedure on synthetic sequences, we now wish to utilize the technique in real scenarios. There is a caveat when using this method, however. As alluded to in Section 6.1.4, transforming between the significance and probability domains is simple when the PDFs are the same shape everywhere as they are in the $\mathcal{M}_0$ case. When the scene is truly static and $\mathcal{M}_0$ is a sufficient model of the background, the FDR procedure can easily be used with the other detection methods presented in Chapter 5. When the background PDFs at each pixel are arbitrary, however, transforming between the two domains is not trivial.

For arbitrary background PDFs, determining the p-score of a given observation requires explicit evaluation of the integral (6.1). Once these scores are determined for every pixel, the FDR procedure can be used to determine a significance threshold p^*. Translating this significance threshold back into a probability threshold, θ, is again not a trivial task.

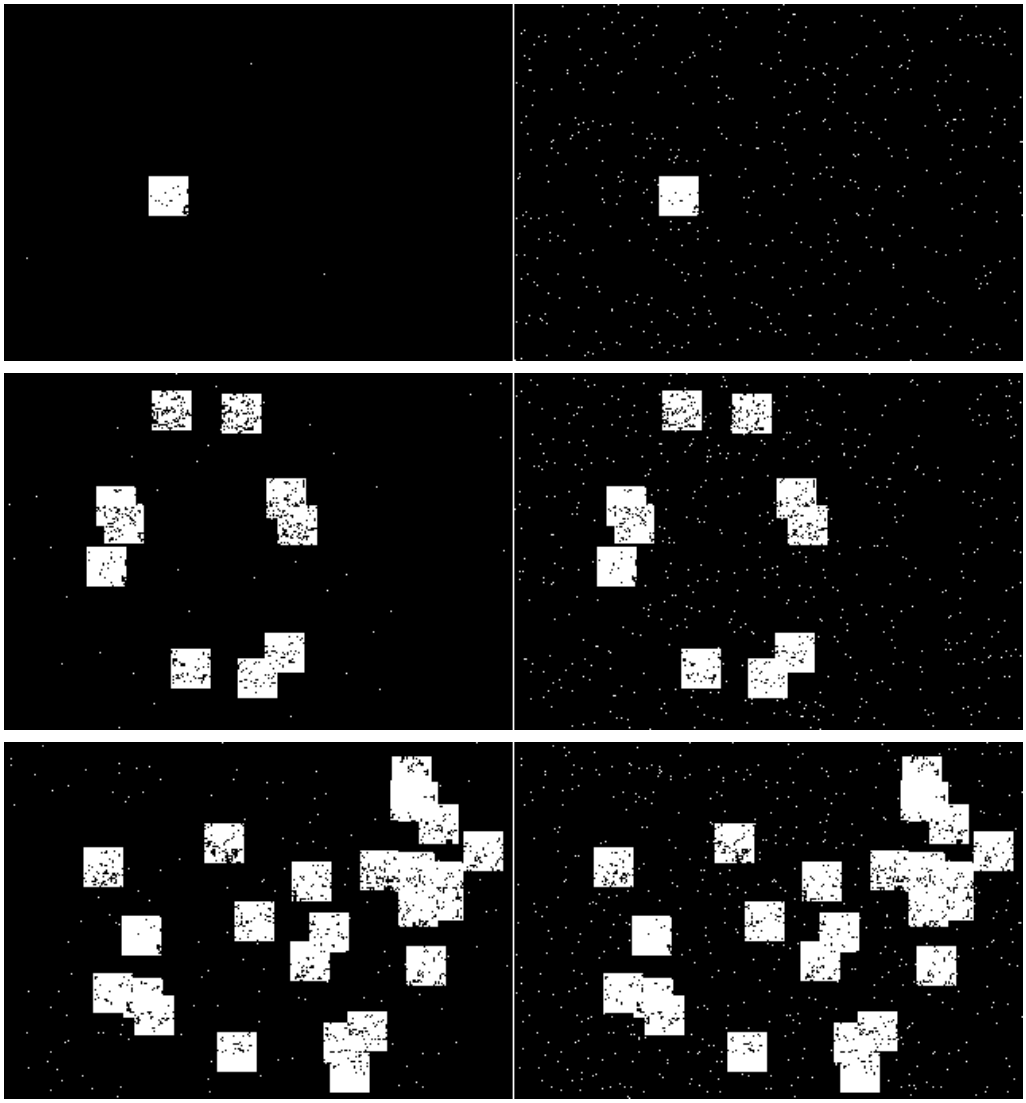

Figure 6.4: Sample detection results with FDR controlling procedure (left images) and FPR controlling procedure (right images). Visually, the improvement is immediate.

Essentially, one must determine the θ that corresponds to the decision region $\mathcal{R}_1$ such that

$$p^* > f(\theta) = \int_{\mathcal{R}_1(\theta)} P_0(y)dy.$$

For a non-parametric PDF, this requires a "water-filling" type of computation at each pixel whereby $\mathcal{R}_1$ is grown until $p^* \leq f(\theta)$. These non-trivial transformations require not only more memory to store the PDFs but also a substantial amount of time to compute (on the order of minutes per frame in our initial MATLAB experimentation). For this reason, extensive results are not provided for the more complex models $\mathcal{M}_1$ and $\mathcal{M}_2$.

An example of the FDR procedure applied to a real sequence is presented in Figures 6.5 and 6.6. The LIT non-parametric background model was used to determine $P_{\mathcal{B}}$ at each pixel and p-scores for each observation in the current image were computed according to (6.1). Notice that, as expected, the number of false positives is lower when the FDR is

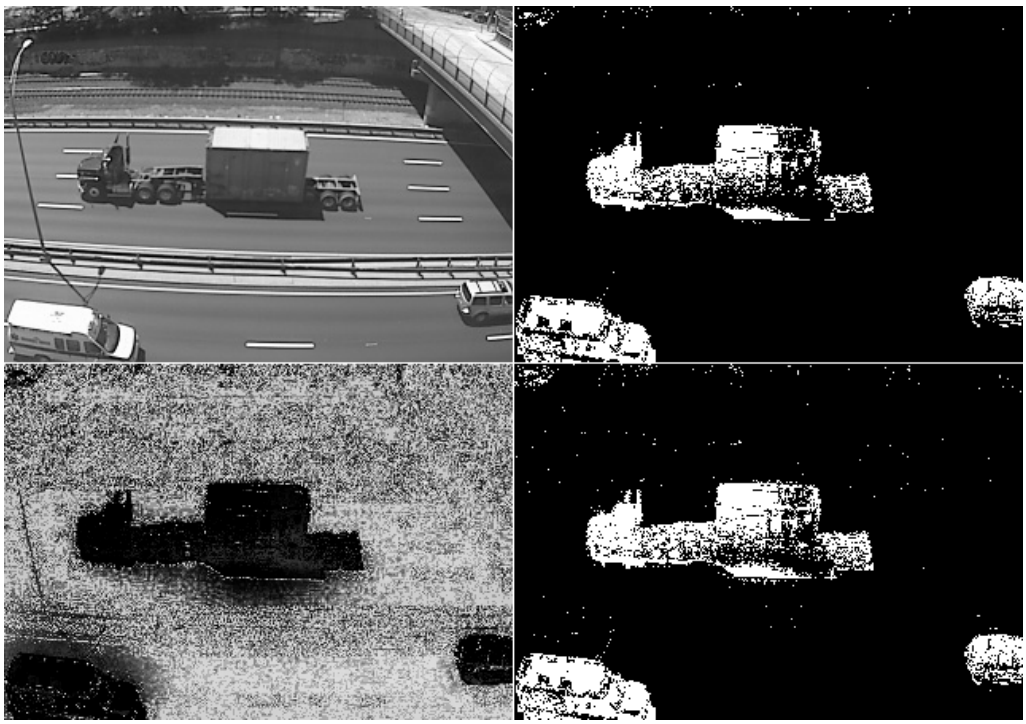

Figure 6.5: Frame from a real image sequence with medium object density. Counter-clockwise from top left: frame; significance image; FPR detection result; FDR detection result. ($\alpha = 1\%$ for both).

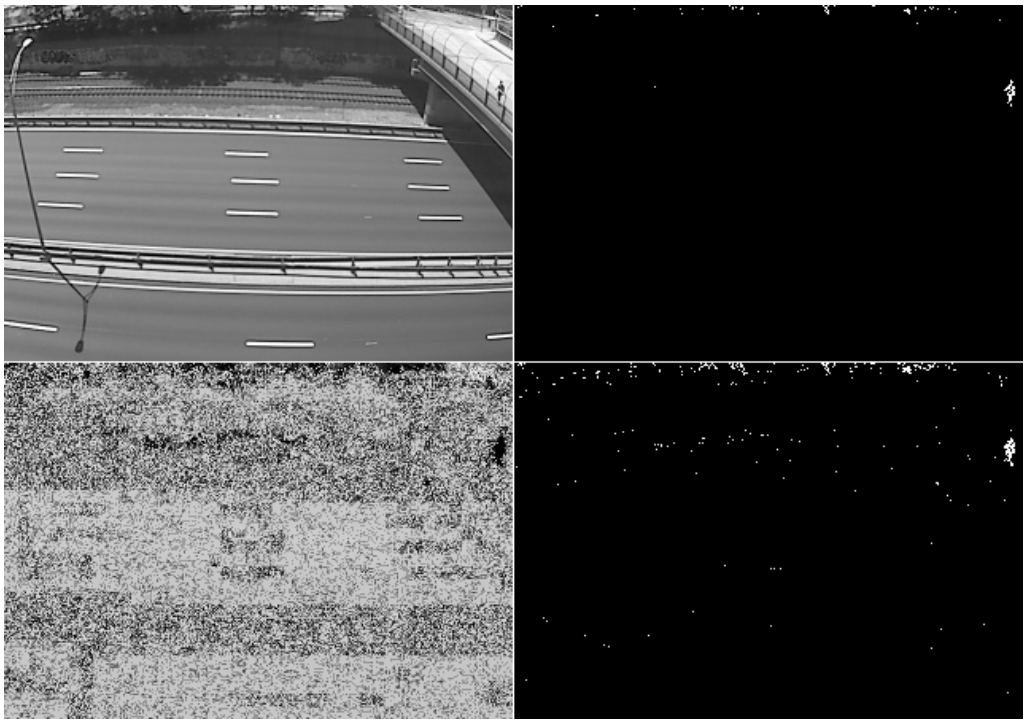

Figure 6.6: Frame from a real image sequence with low object density. Counter-clockwise from top left: frame; significance image; FPR detection result; FDR detection result. ($\alpha = 1\%$ for both).

controlled as opposed to the FPR itself. Additionally, the FDR controlled result improves noticeably when the true density of foreground objects is low whereas the FPR controlled result does not.

Figure 6.7 shows a frame from a testing sequence with a static background and the evolution of the detection process. Only part of the entire frame was used since the top portion contains trees which the background model ($\mathcal{M}_0$) cannot describe well. First, we determined the significance threshold $p^*[k]$ for the current frame (k denoting the time index) by computing p at every pixel according to (6.3) and then applying the FDR procedure. This FDR controlled significance threshold is then transformed to the probability space by (6.4) and an equivalent FDR controlled $\theta^*[k]$ is determined. With this time varying threshold, we apply the foreground and Markov modeling techniques presented in Chapter 5, which adapt the decision process spatially. Figure 6.8 shows how the base detection threshold,

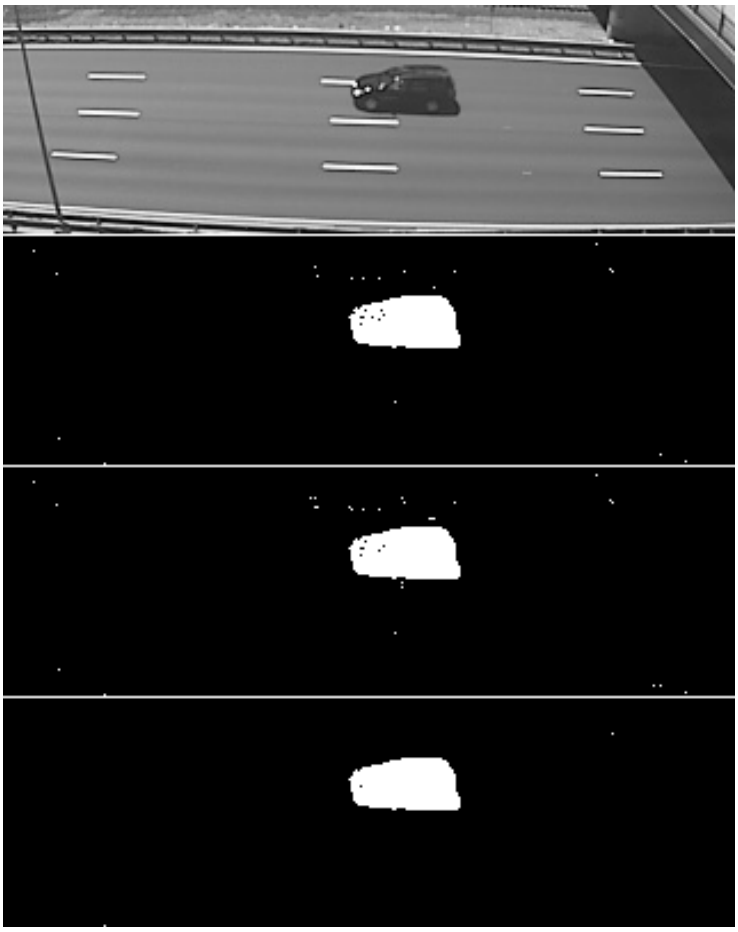

Figure 6.7: A sample frame and detection results after three phases. Top to bottom: thresholding $P_{\mathcal{B}}$ with FDR controlled, time varying $\theta^*[k]$; inclusion of explicit foreground model $P_{\mathcal{F}}$; final detection mask with inclusion of MRF prior.

$\theta^*[k]$ adapts in time to object density.

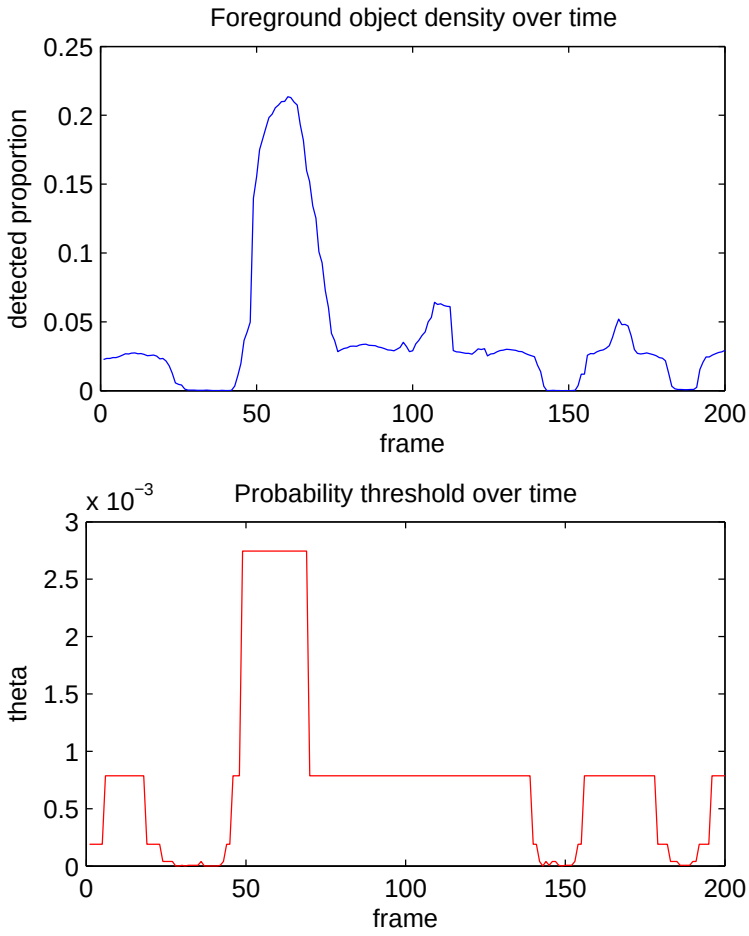

Figure 6.8: The detection threshold, $\theta^*[k]$, is allowed to vary in time when the FDR controlling procedure is used. One can see by comparing the two plots that the threshold adapts to the density of foreground objects.

Chapter 7

Concluding remarks and future research potential

7.1 Discussion of results

In this work, multiple probabilistic modeling methods have been applied to background subtraction and their efficacies have been analyzed and interpreted. With the background subtraction task posed as a classical hypothesis testing problem, we were able to refine estimates for the commonly overlooked parameters - $P_{\mathcal{F}}$ and the *a priori* probabilities - with evidence from the data. Also, by considering the original detection task in the statistical significance domain, we are able to set an initial detection threshold that explicitly controls the error rate. Furthermore, the framework and methods we've presented provide a general basis for further extensions.

Inspired by the work of A. ElGammal *et al.*, non-parametric PDF estimates were utilized for their simplicity and their accurate modeling ability. Three background modeling methods based on locally observed content - background frame, local-in-time, and local-in-space - were presented and contrasted. What we have observed is that the LIT background model is the most accurate of the three for many common scenes, provided that inter-frame misalignment can be handled. Although these specific background models were discussed here, it is clear that an estimate for $P_{\mathcal{B}}$ formulated in different manner will apply equally to our detection methods.

Turning to background detection, two spatially adaptive methods were presented in this thesis. Substantial gains, both visual and numerical, are afforded by incorporating the MRF prior weighting technique. The newly developed foreground modeling procedure has also been shown to be a more effective than fixed thresholding at suppressing misses when

camouflaging is present. Furthermore, by joining the two methods we can achieve superior detection performance in both the quantitative and qualitative senses. Admittedly, the additional gains afforded by incorporating *both* detection techniques are modest compared to those afforded by using the MRF stage alone. However, by incorporating the foreground model at the cost of performing one additional computationally light processing step, the detection performance can be improved, which is encouraging. Overall, our ability to perform reliable background subtraction has been substantially increased by the inclusion of processing steps which are very simple to implement.

Additionally, we have presented a means by which the algorithm designer may achieve predictable performance in terms of error rate, without having to heuristically set a detection threshold. Moreover, by applying the FDR controlling procedure to background subtraction, we have a detector which adapts to the scene over time to further improve performance.

With the three detection techniques presented in this thesis, we have developed a background subtraction algorithm that is adaptive to three different characteristics of the surveilled scene: the overall density of objects over time, the intensity (color) characteristics of ephemeral moving objects, and the local spatial structure of the final detection mask. In comparison to the traditional fixed thresholding detection modality, incorporating these forms of adaptivity has been shown to offer substantial gains.

7.2 Further research areas

Because the detection methods in this thesis have been formulated in general frameworks, many possible extensions to this work are possible.

An interesting study would be to examine the effect of these techniques in higher feature spaces. In this work, only intensity features were considered but the non-parametric modeling techniques easily generalize. It is well known that humans perceive variation in intensity more acutely than color. Most color space transformations and video coding exploit this by separating the luminance from chrominance channels (e.g., YUV, Lab, HSV,

etc.) and quantizing the latter more coarsely. To the eye, luminance (intensity) appears to be one of the most discerning image features. Incorporation of other image attributes may offer better discrimination for object detection.

In (Elgammal et al., 2000; Elgammal et al., 2002), the extension to color space is explicated and demonstrated. Allowing the *color* kernel of the non-parametric PDF to be independent (i.e., vector Gaussian with a diagonal covariance matrix), implementing background modeling is simple. It entails performing a sum of a product of table lookups:

$$P_{\mathcal{B}}(x) = \frac{1}{|\mathcal{M}|} \sum_{[\boldsymbol{m},k]\in\mathcal{M}} \prod_{j=1}^{c} \mathcal{K}_j(x - I_j[\boldsymbol{m},k])$$

where c is the number of color channels, I_j denotes the j^{th} color channel, and each kernel $\mathcal{K}_j$ is zero mean and parameterized by variance σ_j^2.

This concept may be extended further to include other image features such as intensity (color) gradients and edges which may offer greater discrimination. In (Lo and Velastin, 2001), for example, the input images to the background subtraction algorithm are first fed through what the authors dub a "variance filter". This feature extractor computes, for each pixel, the variance of the pixels in a 3×3 window. The claim is that the local variance is invariant to illumination, characterizes edge information, and preserves surface textures.

The newly proposed foreground modeling technique presented in Section 5.1 may be improved in a number of ways. In our formulation, we intentionally exclude any *shape* models specific to particular types of objects. The inclusion of meaningful and general object shape models may be advantageous. As mentioned, in (Wren et al., 1997) and in (Elgammal et al., 2002), the authors are able to successfully grow foreground regions corresponding to body parts by assuming their shape and orientation *a priori*. In the "Wallflower" algorithm, the foreground region growing method proposed is constrained by multiple characteristics of the initial detection mask. One characteristic of note is that the regions to be grown are connected components with at least four pixels. A similar

constraint in our formulation may reduce the exacerbation of speckle noise.

Another possible extension would be to incorporate a higher level scene understanding application such as object tracking. With reliable tracks, one may be able to predict from previous results where a moving object will likely be in the next frame. Provided that the previous segmentations were reasonably accurate, object intensity (color) PDFs based on those segmentations could be used for the next frame. Prior knowledge of the object's expected location and it's intensity (color) distribution would likely assist the current segmentation task.

In Chapter 6, we have demonstrated the power of performing detection in the statistical significance domain. One substantial difficulty with this is that the transformation from the probability and significance domains is generally not trivial and requires a large amount of extra storage space and computation. With the simple background model, closed form expressions for transforming between the two domains were derived. Finding similar closed form transformations for more descriptive background PDFs would be greatly useful.

Additionally, development of similar techniques in the significance domain itself could lead to very powerful detectors which strictly control multiple error rates. The authors of (Benjamini and Hochberg, 1995) have noted that a procedure very similar to the FDR controlling method could be developed which controls the *false rejection rate*: $FRR = \mathbb{E}\{\mathsf{U}/(M-\mathsf{R})\}$. Implicitly, this would require knowledge of the PDF of the alternative hypotheses, which can be estimated in the manner we have described; essentially this significance-like domain would be tied directly to the false negative rate. Alternatively, applying the "domain transformed FDR" (DTFDR) procedure described in (Ermis and Saligrama, 2006) using our explicit models for $P_{\mathcal{F}}$ could alleviate misses due to camouflaging. The DTFDR procedure precedes the conventional FDR procedure with a step that transforms the significance domain so that p-scores under H_0 remain uniformly distributed while p-scores under H_1 become clustered around zero. Again, this would likely require a large amount of computation unless closed form "domain transforms" for arbitrary $P_{\mathcal{F}}$'s can be found.

Appendix A

Image realignment

Using cameras that are not completely stationary exacerbates the already difficult task of performing background subtraction. The cameras used for this work are AXIS 213 pan-tilt-zoom (PTZ) cameras mounted on the roof of a nine-story building (Fig. A.1). The mounting of the camera allows for wind or vibration of the building to shake the camera. The effect is noticeable even when the wind load is small and worsens when zoomed in on a distant scene.

Figure A.1: Roof-mounted PTZ camera used to capture image sequences.

The background models considered in this work are pixel-based, i.e., each background model $P_{\mathcal{B}}(I[\boldsymbol{n}])$ corresponds to a pixel location $\boldsymbol{n}$. When the video camera itself moves, the location of objects in the scene (as projected onto the image coordinates) varies regardless of whether they are moving or stationary. Constant camera shake greatly corrupts background models that do not include local neighbors.

A.1 Camera motion model

Typical camera motion consists of translation and rotation. *Track*, *boom*, and *dolly* designate left-right, up-down, and forward-backward translational motion respectively. *Pan*, *tilt*, and *roll* designate rotation about the vertical, horizontal, and optical axes respectively. Two-dimensional image motion models corresponding to these camera motions are described in (Wang et al., 2002) and (Stiller and Konrad, 1999). The motion models are essentially mapping functions that describe the displacement of all points in the image resulting from camera motion.

A camera undergoing translation, scale (zoom), and rotation is characterized by the *geometric mapping*, which is a special case of the *affine mapping*, and has the form

$$\begin{bmatrix} n_1' \\ n_2' \end{bmatrix} = \begin{bmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} + \begin{bmatrix} c_3 \\ c_4 \end{bmatrix}$$

where $[n_1\ n_2]^T$ and $[n_1'\ n_2']^T$ denote image coordinates before and after the motions.

The motion that our cameras undergo can be described by a combination of translations and small rotations. Also, the imaged scene is far away from the camera. Because of these two important factors, the affine model can be simplified and we can assume that the imaged scene is displaced uniformly (Wang et al., 2002). That is to say, there is no rotation or scale relationship between two images, only translation. With no other motion in the scene, the relationship between any two frames of the video is then simply $I'[\boldsymbol{n}] = I[\boldsymbol{n} - \boldsymbol{d}_0]$.

A.2 Phase correlation

In order to realign images captured from the shaky camera, one must find the displacement parameter $\boldsymbol{d}_0$. One relatively fast way to do this is to use the so-called phase correlation (PC). This technique is used for block matching. It mitigates exhaustive searching using the correlation criterion as opposed to a pixel-based error criterion (Konrad, 2005).

The correlation between two images is denoted

$$C[\mathbf{d}] = \sum_{\mathbf{n}} I_1[\mathbf{n}] I_2[\mathbf{n} - \mathbf{d}] \tag{A.1}$$

To clarify, the correlation is defined over $\boldsymbol{d} \in \Gamma \subseteq \Lambda \subset \mathbb{R}^2$ which is a discrete space domain of delays or lags, whereas $\mathbf{n} \in \Lambda$ is the original image domain. The sampling lattice Γ may be identically equal to Λ or it may be denser by an integral factor (see Figure A.2).

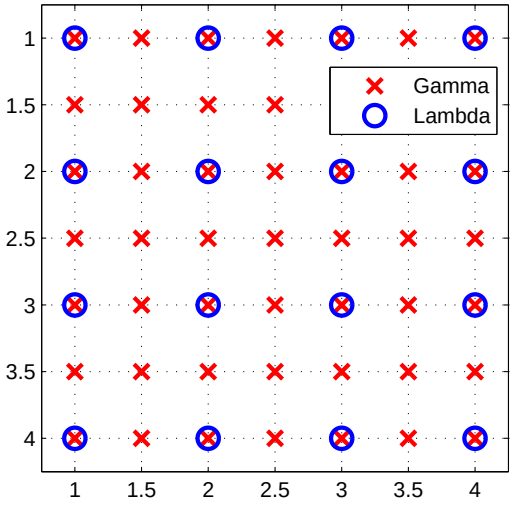

Figure A.2: Example image pixel domain lattice, Λ, and correlation lag domain lattice, Γ, with a factor of 2 scale.

The correlation has a very convenient form in the discrete 2D Fourier domain:

$$\widehat{C}[\boldsymbol{u}] = \widehat{I}_1[\boldsymbol{u}] \widehat{I}_2^*[\boldsymbol{u}] \tag{A.2}$$

where the hat denotes the 2-D discrete Fourier transform (DFT) version of the signals and the star superscript denotes the complex conjugate. This relationship can be shown rather

easily using Fourier transform properties.

$$\widehat{C}[\boldsymbol{u}] = \mathrm{DFT}\left\{C[\boldsymbol{d}]\right\} \tag{A.3a}$$

$$= \mathrm{DFT}\left\{\sum_{\boldsymbol{n}} I_1[\boldsymbol{n}] I_2[\boldsymbol{n}-\boldsymbol{d}]\right\} \tag{A.3b}$$

$$= \mathrm{DFT}\left\{\sum_{\boldsymbol{n}} I_1[\boldsymbol{n}] I_2[-(\boldsymbol{d}-\boldsymbol{n})]\right\} \tag{A.3c}$$

$$= \mathrm{DFT}\left\{I_1[\boldsymbol{d}] * I_2[-(\boldsymbol{d}-\boldsymbol{n})]\right\} \tag{A.3d}$$

$$= \widehat{I}_1[\boldsymbol{u}]\widehat{I}_2^*[\boldsymbol{u}] \quad \blacksquare \tag{A.3e}$$

The star operator on line (A.3d) indicates 2D convolution. Normalizing this function by its magnitude and taking the inverse discrete Fourier transform (IDFT) leads to a normalized correlation function:

$$\begin{aligned}\Psi_{1,2}[\mathbf{d}] &= \mathrm{IDFT}\left\{\frac{\widehat{C}[\boldsymbol{u}]}{|\widehat{C}[\boldsymbol{u}]|}\right\} \\ &= \mathrm{IDFT}\left\{\frac{\widehat{I}_1[\boldsymbol{u}]\widehat{I}_2^*[\boldsymbol{u}]}{|\widehat{I}_1[\boldsymbol{u}]\widehat{I}_2^*[\boldsymbol{u}]|}\right\}.\end{aligned} \tag{A.4}$$

In the special case of global translation (i.e., $I_1[\boldsymbol{n}] = I_2[\boldsymbol{n}-\boldsymbol{d}_0]$), the normalized correlation function becomes a Kronecker delta function at the location of the displacement vector. This can be easily seen by manipulating the argument of the IDFT in expression (A.4).

$$\frac{\widehat{I}_1[\boldsymbol{u}]\widehat{I}_2^*[\boldsymbol{u}]}{|\widehat{I}_1[\boldsymbol{u}]\widehat{I}_2^*[\boldsymbol{u}]|} = \frac{|\widehat{I}_1[\boldsymbol{u}]||\widehat{I}_2[\boldsymbol{u}]| \exp\left(j\left(\angle\widehat{I}_1[\boldsymbol{u}] - \angle\widehat{I}_2[\boldsymbol{u}]\right)\right)}{|\widehat{I}_1[\boldsymbol{u}]||\widehat{I}_2[\boldsymbol{u}]|} \tag{A.5a}$$

$$= \exp\left(j\left(\angle\widehat{I}_1[\boldsymbol{u}] - \angle\widehat{I}_2[\boldsymbol{u}]\right)\right) \tag{A.5b}$$

$$= \exp\left(j\left(\angle\widehat{I}_1[\boldsymbol{u}] - \angle\widehat{I}_1[\boldsymbol{u}] - 2\pi\boldsymbol{u}^T\boldsymbol{d}_0\right)\right) \tag{A.5c}$$

$$= \exp\left(-j2\pi\boldsymbol{u}^T\boldsymbol{d}_0\right) \quad \blacksquare \tag{A.5d}$$

Thus the inverse DFT of $\widehat{\Psi}_{1,2}$ produces $\delta[\boldsymbol{d}-\boldsymbol{d}_0]$.

Since there is generally other, non-global movement in the scene by nature of the application, less dominant peaks will arise in the correlation surface. Edge effects will also

distort the estimate. To determine the displacement, one may take the following steps:

1. Choose one frame as I_0, the frame to which we shall realign subsequent frames.
2. Take equally sized blocks (preferably square with 2^n-length sides, e.g., 64×64) from I_0 and the current frame I_k, and take 2-D FFT of each.
3. Compute $\Psi_{0,k}$ using 2-D inverse FFT as in (A.4).
4. Search for coordinates of most dominant peak in $\Psi_{0,k}$.

Subpixel registration accuracy to $1/s^{th}$-pixel can be achieved by zero padding $\widehat{\Psi}_{0,k}$ to $s \times s$ its original size before taking the inverse FFT, effectively interpolating the correlation surface. In itself, determining the displacement with subpixel accuracy clearly requires more computation. Additionally, realigning I_k with a fractional offset requires interpolation in the space domain which in addition to requiring still more computation has a lowpass effect on the image. For most of the methods discussed in this thesis, full pixel accuracy is adequate (i.e., $\Gamma = \Lambda$).

The current MATLAB implementation of the PC realignment stage requires less than one tenth of a second per frame with single pixel resolution (i.e., no interpolation) and with a block (FFT) size of 64×64.

References

Aach, T. and Kaup, A. (1995). Bayesian algorithms for adaptive change detection in images sequences using markov random fields. *Signal Processing: Image Communication*, 7:147–160.

Aach, T., Kaup, A., and Mester, R. (1993). Statistical model-based change detection in moving video. *Signal Processing*, 31:165–180.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B*, 57(1):289–300.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, B 48:259–279.

Butler, D., Sridharan, S., and V. Bove Jr. (2003). Real-time adaptive background segmentation. *Proceedings of the International Conference on Multimedia and Expo*, 3:III–341–4.

Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342.

Cutler, R. and Davis, L. (1998). View-based detection and analysis of periodic motion. *Proceedings of the International Conference on Pattern Recognition*, 1.

Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L. (2002). Background and foreground modeling using nonparametric kernel density for visual surveillance. *Proceedings of the IEEE*, 90:1151–1163.

Elgammal, A., Harwood, D., and Davis, L. (2000). Non-parametric model for background subtraction. In *Proceedings of the European Conference on Computer Vision*.

Ermis, E. and Saligrama, V. (Submitted Dec 2006). Distributed detection in sensor networks with limited sensing range. *IEEE Transactions on Information Theory*.

Friedman, N. and Russell, S. (1997). Image segmentation in video sequences: A probabilistic approach. In *Annual Conference on Uncertainty in Artificial Intelligence*, pages 175–181.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Jodoin, P.-M., Mignotte, M., and Konrad, J. (2006a). Background subtraction framework based on local spatial distributions. In *Proceedings of the International Conference on Image Analysis and Recognition*, pages 370–380.

Jodoin, P.-M., Mignotte, M., and Konrad, J. (2006b). Light and fast statistical motion detection method based on ergodic model. In *Proceedings of the IEEE International Conference on Image Processing*.

Konrad, J. (2005). Motion detection and estimation. In Bovik, A., editor, *Handbook of Image and Video Processing, 2nd Edition*, chapter 3.10, pages 253–274. Academic Press.

Lehmann, E. L. (1986). *Testing Statistical Hypotheses*. John Wiley and Sons, 3rd edition.

Lo, B. P. L. and Velastin, S. A. (2001). A congestion detection system for underground platforms. *Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 158–161.

McFarlane, N. and Schofield, C. (1995). Segmentation and tracking of piglets in images. *Machine Vision Applications*, 8(3):187–193.

Migdal, J. and Grimson, W. E. L. (2005). Background subtraction using markov thresholds. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, volume 2, pages 58–65, Washington, DC, USA. IEEE Computer Society.

Moghaddam, B. and Pentland, A. (Jul 1997). Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710.

Noriega, P., Bascle, B., and Bernier, O. (2006). Local kernel color histograms for background substraction. *International Conference on Computer Vision Theory and Applications*.

Oliver, N., Rosario, B., and Pentland, A. (2000). A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843.

Radke, R. (2005). Image change detection algorithms: A systematic survey.

Stauffer, C. and Grimson, E. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757.

Stiller, C. and Konrad, J. (1999). Estimating motion in image sequences: A tutorial on modeling and computation of 2D motion. *IEEE Signal Processing Magazine*, 16(4):70–91.

Theodoridis, S. and Koutroubas, K. (2006). *Pattern Recognition*. Academic Press, 3rd edition.

Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 255–261.

Wang, Y., Ostermann, J., and Zhang, Y.-Q. (2002). *Video Processing and Communications*. Prentice Hall.

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

J. Mike McHugh

35 Long Avenue, Allston, MA
jmmchugh@gmail.com

Education

Boston University, College of Engineering - Boston, MA
M.S. Electrical and Computer Engineering - January 2008
GPA 3.9, Concentration - Signal Processing and Communications
Thesis: "Probabilistic Methods for Adaptive Background Subtraction."
Coursework: Advanced Digital Signal Processing, Multidimensional Signal and Image Processing, Statistical Pattern Recognition, Sensor Networks, Stochastic Processes, Information Theory

Worcester Polytechnic Institute - Worcester, MA
B.S. Electrical and Computer Engineering - May 2006
GPA 3.8, High Distinction, Salisbury Prize
Coursework: Digital Signal Processing Applications, Communication Systems Engineering, Embedded Computer Systems, C++ Programming Techniques, Advanced Logic Design, Microelectronic Circuits

Recent Work

Boston University, ECE Department
Research Assistant, Advisor Prof. J. Konrad
Summer 2007 through Fall 2007
Primary work in moving object detection algorithms for video surveillance. Researched various statistical methods for background modeling and subtraction. Developed a foreground object modeling procedure to enhance detection results. Most recent research in video copy detection methods.
Unpublished Technical Paper: "Motion Detection With False Discovery Rate Control" Submitted to IEEE 2008 International Conference on Image Processing

Mercury Computer Systems, Hardware Group
Electrical Engineering Co-op
March 2006 through May 2007
Lead electrical engineer for Serial RapidIO loopback test card, PCI add-in ATA Controller card, and Cell Workstation test card. Duties included circuit design, schematic capture, PCB layout oversight, design validation and testing, and documentation.

Boston University, ECE Department
Grader for Multidimensional Signal and Image Processing Class
Fall 2007

Worcester Polytechnic Institute, ECE Department
Tutor for Digital Signal Processing Applications Class
Spring 2006

Academic Projects

Information Theory Course Project
Spring 2007
"Classification Using Approximate Information Theoretic Measures"
Investigated use of approximate information theoretic quantities, determined using universal data compressors, as distance measures for classification. Applied methods to text and speech signals for author/speaker recognition.

Advanced Digital Signal Processing Course Project
Spring 2007
"Digital Sampling Rate Alteration Using Polyphase Decomposition"
Researched and discussed efficient sampling rate alteration techniques utilizing multi-level polyphase decomposition.

Statistical Pattern Recognition Course Project
Fall 2006
"Covariance Matrix as a Region Descriptor for Object Detection in Images"
Evaluated a novel image matching technique which uses covariance matrix of features (color, edges, etc.) as a region descriptor. Reported performance for various feature and image sets.

ECE Capstone Design Project
Bose Corporation & Signal Processing and Information Networking Lab at WPI
Academic year 2005-2006
"Many-to-Many Wireless Music Distribution System"
Developed a wireless music distribution system that utilizes an 802.11 link to stream audio between multiple common home audio devices. Designed multi-process server/client and UI software in C++ to run on Linux OS.

Skills

Languages: MATLAB, C/C#/C++, nesC, assembly (PIC, TI DSPs, x86)
Operating Systems: Microsoft Windows XP, Linux/UNIX, tinyOS
Software: Microsoft Visual Studio, Code Composer Studio, Viewdraw, Microsoft Office
Technical: Digital High-Speed Oscilloscope, Function Generator, Digital Multimeter

Professional Organization Memberships

IEEE, Graduate Student Member
Tau Beta Pi, National Engineering Honor Society
Eta Kappa Nu, National ECE Honor Society