Report Number: WUCSE-2004-25

2004-05-11

Modeling Local Video Statistics for Anomaly Detection

Roman Garnett

This paper promotes a probabilistic approach for building models of local video statistics for use in background subtraction schemes. By shifting into a probabilistic framework, additional analytical tools become available for the creation and evaluation of these models. This paper continues to suggest the use of nonparametric statistical methods for measuring the quality of efficient local spatio-temporal models of video background distributions. Beginning with the familiar relative entropy distance between probability distributions, we create a new distance measure that can be used to quantitatively measure the quality of a probabilistic background model.
... Read complete abstract on page 2.

Modeling Local Video Statistics for Anomaly Detection

Roman Garnett

Complete Abstract:

This paper promotes a probabilistic approach for building models of local video statistics for use in background subtraction schemes. By shifting into a probabilistic framework, additional analytical tools become available for the creation and evaluation of these models. This paper continues to suggest the use of nonparametric statistical methods for measuring the quality of efficient local spatio-temporal models of video background distributions. Beginning with the familiar relative entropy distance between probability distributions, we create a new distance measure that can be used to quantitatively measure the quality of a probabilistic background model.

Department of Computer Science & Engineering

Washington University in St. Louis
School of Engineering & Applied Science

2004-25

Modeling Local Video Statistics for Anomaly Detection, Master's Thesis, May 2004

Authors: Garnett, Roman

May 11, 2004

Abstract: This paper promotes a probabilistic approach for building models of local video statistics for use in background subtraction schemes. By shifting into a probabilistic framework, additional analytical tools become available for the creation and evaluation of these models. This paper continues to suggest the use of nonparametric statistical methods for measuring the quality of efficient local spatio-temporal models of video background distributions. Beginning with the familiar relative entropy distance between probability distributions, we create a new distance measure that can be used to quantitatively measure the quality of a probabilistic background model.

WASHINGTON UNIVERSITY

SEVER INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MODELING LOCAL VIDEO STATISTICS FOR ANOMALY DETECTION

by

Roman Garnett

Prepared under the direction of Professor Robert Pless

A thesis presented to the Sever Institute of
Washington University in partial fulfillment
of the requirements for the degree of

Master of Science

May, 2004

Saint Louis, Missouri

WASHINGTON UNIVERSITY
SEVER INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ABSTRACT

MODELING LOCAL VIDEO STATISTICS FOR ANOMALY DETECTION

by Roman Garnett

ADVISOR: Professor Robert Pless

May, 2004
Saint Louis, Missouri

This paper promotes a probabilistic approach for building models of local video statistics for use in background subtraction schemes. By shifting into a probabilistic framework, additional analytical tools become available for the creation and evaluation of these models.

This paper continues to suggest the use of nonparametric statistical methods for measuring the quality of efficient local spatio-temporal models of video background distributions. Beginning with the familiar relative entropy distance between probability distributions, we create a new distance measure that can be used to quantitatively measure the quality of a probabilistic background model.

Contents

List of Tables

List of Figures

Acknowledgments

Thanks to everyone who helped me finish this in a punctual manner.

Roman Garnett

Washington University in Saint Louis
May 2004

Chapter 1

Introduction

There is an increasing need for automated video surveillance systems. Although surveillance cameras are widespread, it is difficult to find sufficient human resources to monitor and interpret the enormous amount of data they produce. As a result, surveillance data are often used to identify criminal or otherwise anomalous activity only after it has occurred. By combining surveillance cameras with computer systems capable of automatically monitoring and interpreting the video data in real-time, we can detect anomalous activity as it occurs.

An important component of video surveillance systems is *background subtraction*, the identification of background information that can be safely ignored during the search for foreground anomalies. With a computationally efficient background subtraction component, a surveillance algorithm can spend more resources on identifying and tracking anomalies.

The complexity of different background subtraction systems varies wildly. In the entertainment industry, for example, blue screens are used to give the background a known and constant intensity that is easy to identify and ignore. Unfortunately, real data rarely behave so nicely. Consequently, more specialized and tolerant background models have been developed, some of which can even ignore consistent motion (like normal movement in traffic scenes) when it occurs in the background.

Although there are already some useful background maintenance systems in existence (see, for example, [18]), they suffer from several significant problems. One of these is their lack of a strong theoretical basis. The design of these systems is largely *ad hoc*, and methods are motivated by the quality of their results rather than by the theoretical soundness of their definitions. In particular, little effort has been made to quantitatively measure how well a particular background model reflects the true background distribution.

In this paper we try to close this theoretical gap by promoting the use of nonparametric statistics for measuring background model quality.

1.1 Previous Methods

The general framework of most automated surveillance systems is simple. A collection of training video data are used to build a model of the background. The system then compares new data to the model and identifies which pixels fit it well. Any remaining pixels are then further processed and tracked, if necessary.

Previous work on background subtraction has focused on calculating the expected intensity of a pixel [6] and on identifying and classifying consistent motion within a scene [5, 20, 11]. A good overview of these and other methods can be found in [18], in which the authors give several guidelines for designing successful background maintenance systems.

In general, these methods can operate on several different scales—local methods that model the background distribution at each image pixel [12, 21, 4, 5], methods that operate on image regions [9], or methods that operate on entire frames [10]. Some more complicated methods use multiscale detectors that operate at all three levels [18].

In [12], the authors describe several potentially useful pixel-level background models and include instructions for generating the models, estimating their parameters online, and scoring new measurements.

1.1.1 Problems with Previous Methods

These previously proposed methods all lack a sound theoretical basis. Models are generally measured in terms of their use as dichotomous classifiers. Their quality is judged by a count of the number of falsely classified objects in test sequences or by an investigation of their receiver-operating characteristics over a large range of possible thresholds.

Neither of these procedures attempts in any way to determine just how well a generated model represents the true background model. Rather, the quality of a method is determined solely by its results on a small number of examples. Although there is merit to these observations, it is unclear whether such an approach truly measures a potential model's ability to model many different background distributions.

We discuss these problems in more detail in Chapter 2.

1.2 Overview of Remainder

We begin by establishing a standard notation to be used throughout the paper in Chapter 2. We proceed in that chapter to review previous methods and to discuss the problems that they suffer. In Chapter 3, we promote a probabilistic approach to background modeling instead of the classifier approach traditionally used. In Chapter 4, we use nonparametric density estimation to help calculate a new distance measure on probability density functions. Finally, in Chapter 5, we support the measure defined with experimental results.

A review of kernel density estimation is given in Appendix A.

Chapter 2

Notation and Review of Relevant Previous Work

We begin by defining the notation used throughout this paper.

2.1 Video Notation

We will consider a video to be a sequence of images. Let V represent an N-frame video, each frame of which is an $n \times m$ image. We will then write $V = \{I(t)\}_{t=1}^{N}$, where $I(t)$ represents the t^{th} frame. To represent the intensity of the component image $I(t)$ at the pixel (x, y), we write $I(x, y, t)$. This intensity can either be a scalar (for monochrome videos) or a vector (for color videos).

Throughout this paper, it will often be helpful to think of a video as being a three-dimensional function that describes how the intensity of a video changes in the spatial domain over time. We may then consider this function's spatial and temporal derivatives. Suppose (x, y) is a pixel of interest. We will use $I_x(x, y, t)$, $I_y(x, y, t)$, and $I_t(x, y, t)$ to represent approximations of the spatial derivative in the horizontal direction, the spatial derivative in the vertical direction, and the temporal derivative at (x, y) at time t. We will often drop the (x, y, t) parameters when unnecessary.

2.2 Probability Notation

We will frequently discuss several common probability density functions (PDFs). Suppose that Ω is a measurable state space, that $\mathscr{F} \subseteq \mathscr{P}(\Omega)$ is a set of measurable subsets of Ω, and that P is a probability measure on Ω, that is, a measure with $P(\Omega) = 1$. If $\Delta \in \mathscr{F}$, we can create the uniform distribution on Δ. We will denote this distribution as $U(\Delta)$.

If we are working in the familiar state space $\mathbb{R}^n$, we encounter several important probability density functions. Perhaps the most important is the multivariate normal distribution. Let $\mu \in \mathbb{R}^n$ be a vector and $\mathbf{\Sigma}$ be a symmetric, positive-definite $n \times n$ matrix. We will use $N(\mu, \mathbf{\Sigma})$ to represent the multivariate normal distribution with mean μ and covariance $\mathbf{\Sigma}$.

Throughout the paper we will assume that any covariance matrix $\mathbf{\Sigma}$ is symmetric and positive-definite.

2.3 Review of Previous Work

Many background maintenance schemes have been developed. In [18], the authors examine the performance of 10 such schemes. They describe several problematic situations of which background maintenance systems should be aware and compare each method's performance in response to these problematic scenarios. In the end, they conclude that none of the schemes is able to correctly handle all of the problem situations they identify, but with the insight gained from their investigation they propose five principles by which all background maintenance schemes should abide. We list these principles below, as stated in [18]:

- Semantic differentiation of objects should not be handled by the background maintenance module.
- Background subtraction should segment objects of interest when they first appear (or reappear) in a scene.

- An appropriate pixel-level stationarity criterion should be defined. Pixels that satisfy this criterion are declared background and ignored.
- The background model must adapt to both sudden and gradual changes in the background.
- Background models should take into account changes at differing spatial scales.

2.4 Pixel-Level Spatio–Temporal Models

In [12], a special, robust class of background models is considered. A video sequence is interpreted as a three-dimensional intensity function on which local spatial and temporal derivatives are defined. At each pixel (x, y) in the component images, an independent background model is constructed. The basis for these models is the 4-vectors $[I, I_x, I_y, I_t](x, y, t)$.

After the background models have been constructed, new data may be scored using a function $f_{(x,y)}(I, I_x, I_y, I_t)$ that represents the negative log-likelihood that the measurement $[I, I_x, I_y, I_t]$ came from the background model generated at the pixel (x, y). If a measurement has a score below some set threshold, it is considered to come from the background; if its score exceeds the threshold, it is marked for additional processing.

In this paper, we focus on models of the type described in [12]. With this framework, each model is completely described by its measurement definition, scoring function, and parameter estimation method. The most important models are those for which parameters can be estimated online. If the system can update model parameters online, it can adapt to changing background conditions in real-time. With an online scheme, the background model and scoring function can potentially abide by the first four of the principles identified above, although they will continue to operate on a single scale. It would be trivial to implement a multiscale approach, however, by first subsampling the component images and then building local models

on the subsampled video. In such a scheme, these background subtraction models can be expected to satisfy all five of the criteria, assuming a reasonable choice of model and scoring function. For this reason, we study these models exclusively.

We briefly present the three background models described by [12] that we consider in this paper. For each model we enumerate the necessary measurements needed to build the model, the parameters of the model, and the score used to measure how well a new measurement fits into the model. Descriptions of how the model parameters may be estimated, including online methods if available, may be found in [12].

2.4.1 Background of Constant Intensity

If the camera is fixed and the background can be expected to stay relatively constant, we can model the background as a single static image that may be easily identified and ignored. Note that such a model requires that all motion be considered anomalous.

- **Measurement used**: The required measurement is the intensity I.
- **Score**: If the background is assumed to have intensity I', we score an observed intensity I^* by simply taking the squared L^2 distance, $||I' - I^*||^2$.

2.4.2 Background with Gaussian Distribution in (I_x, I_y, I_t) Space

If a scene contains motion that should be considered part of the background, more tolerant models are required. One solution is to model measurements in the (I_x, I_y, I_t) space with a single multivariate Gaussian distribution.

- **Measurements used**: The required measurements are the intensity derivatives, I_x, I_y, and I_t.
- **Parameters**: The parameters of this model are the mean μ and covariance matrix $\mathbf{\Sigma}$.

- **Score**: If background measurements are assumed to come from a multivariate Gaussian distribution $G(\mu, \mathbf{\Sigma})$, the negative log-likelihood that a given measurement $\mathbf{m} = (I_x, I_y, I_t)$ comes from $N(\mu, \mathbf{\Sigma})$ is simply the squared Mahalanobis distance, $(\mathbf{m} - \mu)^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{m} - \mu)$.

2.4.3 Background with A Mixture of Gaussian Distributions

When a single Gaussian is insufficient to model the distribution of (I_x, I_y, I_t) values, a finite mixture of Gaussians may be used instead.

- **Measurements used**: The required measurements are the intensity derivatives, I_x, I_y, and I_t.

- **Parameters**: Suppose the mixture model contains k Gaussians for some $k \in \mathbb{N}$. The parameters of this model are then k mean values $\{\mu_\mathbf{i}\}_{i=1}^{k}$, k covariance matrices $\{\mathbf{\Sigma_i}\}_{i=1}^{k}$, and k scaling factors $\{p_i\}_{i=1}^{k}$, with $0 \leq p_i \leq 1$ and $\sum_{i=1}^{k} p_i = 1$.

- **Score**: If background measurements are assumed to come from a weighted sum of multivariate Gaussian distributions $\{p_i \cdot N(\mu_\mathbf{i}, \mathbf{\Sigma_i})\}_{i=1}^{k}$, the negative log-likelihood that a given measurement $\mathbf{m} = (I_x, I_y, I_t)$ comes from this distribution is a weighted sum of the k squared Mahalanobis distances:

$$\sum_{i=1}^{k} p_i \cdot ((\mathbf{m} - \mu_\mathbf{i})^{\mathrm{T}} \mathbf{\Sigma_i}^{-1} (\mathbf{m} - \mu_\mathbf{i}))$$

There are several methods available for building such a mixture model. A widely used algorithm is expectation maximization, which uses an iterative process to find the best-fitting mixture of Gaussians for a particular dataset [15]. Although expectation maximization can be expected to perform relatively well, the parameters must be updated and calculated offline; therefore, expectation maximization cannot be used when online methods are required.

In [12], the authors consider the finite mixture model for completeness but conclude that "there is no natural method for an incremental EM solution which fits

the streaming video processing model and does not require maintaining a history of all prior data points." One possible solution is to use the adaptive mixture method proposed in [13], which uses a data-driven approach to estimate the parameters of an underlying mixture model. In our simulations this method has proven to be almost as effective as expectation maximization while operating online.

2.5 Receiver-Operating Characteristics as Measures of Quality

As a first attempt at measuring the quality of these local models, Pless *et al.* suggest using receiver-operating characteristic (ROC) plots. These plots attempt to show the general performance of a classifier over the range of its possible threshold values.

Suppose we have a classifier C that accepts a range of real values $[a, b]$ ($a,\ b \in \mathbb{R}$) as its threshold. The ROC plot for C is created by measuring its sensitivity (the portion of true positives correctly classified) and specificity (the portion of true negatives correctly classified) for large number of sample thresholds $T \in [a, b]$. Let s_T represent the sensitivity of the classifier C with threshold T, and let σ_T represent its specificity. For each threshold T, we plot s_T (on the y-axis) against $(1 - \sigma_T)$ on the x-axis. The resulting parameterized curve is the ROC curve for C. To clarify this space, the classifier that always answers "yes" is identified by the point $(1, 1)$ in this space, the classifier that always answers "no" by the point $(0, 0)$, and a perfect classifier by the point $(1, 0)$.

The quality of a particular classifier can be inferred from its ROC plot in several ways. A common approach is to measure the area under the ROC curve. The ROC plot of a perfectly random classifier has slope 1 and intersects both of the points $(0, 0)$ and $(1, 1)$. Its integral is therefore $\frac{1}{2}$, and so any potentially useful classifier should have an integral higher than this. Another popular quality measure is the "distance" between the curve and the perfect classifier point $(1, 0)$. The closest point on the ROC curve of a random classifier is the point $(\frac{1}{2}, \frac{1}{2})$, with distance $\frac{\sqrt{2}}{2}$.

2.5.1 Problems with ROC Plots

The ROC curve approach has several weaknesses. One significant problem is that once an ROC plot is generated, it is impossible to infer the amount or nature of the data considered when creating the plot—information that is clearly important for ascertaining the relevance of the curve.

Another problem concerns threshold selection. Many different thresholds are used to generate an ROC plot. From these plots, one can infer to some extent the general behavior of the classifier and can perhaps gain basic insight into which values may be good thresholds. These plots can vary widely from one application of the classifier to another, though, and the optimal threshold may drastically change with the situation.

Once the classifier is being used on real data, the user cannot know what the relevant ROCs are and may not be able to choose the optimal, or perhaps even a useful, threshold. These reservations regarding threshold choice suggest that a dichotomous classifier may not always be the best choice for video surveillance systems. Instead, we suggest that probability density functions be used to model background distributions, from which useful probabilistic measures may be derived.

Chapter 3

Moving Towards a Probabilistic Model

Pixels in surveillance video data can be capturing data from either the background or the foreground. For this reason, dichotomous classifiers seem to be the natural choice for solving the background subtraction problem. On the other hand, we can understand the classifier models described above as operating in two phases. First, the system builds some parametric model that is assumed to represent the true background distribution. Second, the system gives each new measurement a score that measures how well the measurement fits in the model. The magnitude of this score can, in some sense, indicate the likelihood that the measurement came from the true background distribution—small values suggest that the measurement is likely to have come from the background, whereas large values suggest that the measurement is anomalous. Although this is fairly easy to use in an automatic system, each classifier has a different range of possible scores. Further, the degree to which the magnitude of a measurement's score indicates its likelihood of occurring varies from classifier to classifier, making their use somewhat difficult.

As an alternative to this approach, we can model the true background distribution itself. Instead of merely characterizing the distribution and thresholding distance

scores from some *ad hoc* model, we model a background distribution with a probability density function f. This PDF can lie within any number of finite-dimensional spaces, but for our purposes the most important such space will be the (I_x, I_y, I_t) space used in the three models described in the previous chapter. Then, given a new observation, we can approximate the *probability* that the measurement came from the estimated background distribution. The problem of generating background models, then, becomes a question of probability density estimation.

There are several advantages to this probabilistic approach. It is backwards-compatible with the classifier approach. We may turn any probability density estimate of the background distribution into a classifier in a straightforward, universal way: the user selects a value $p \in [0, 1]$, and any measurement having probability less than p of being in the background distribution is marked for further investigation. This idea is simple, universal, and independent of the ROCs of a classifier.

Additionally, estimating the probability that a measurement came from the background distribution can be useful for presenting surveillance data after it has been scored. For example, the system could use this probability data to color foreground pixels according to their probability of not having come from the background distribution.

Finally, this probabilistic approach is useful because it permits a wealth of statistical methods, both parametric and nonparametric, to become available to us for analyzing and evaluating background models.

Many of the existing classifier-based methods can be easily recast in terms of this probabilistic approach. The three methods presented in the previous chapter, for example, all have simple implicit probability density function representations, which we present in the next section. Almost all of these classifier-based methods, however, rely on parametric estimates of the data. This is probably not the best approach, since true video data is not likely to be distributed as any sort of natural parametric density function.

By taking a probabilistic standpoint to the problem of background modeling, we are able to apply nonparametric probability density estimation techniques that make no assumptions about the distribution of the underlying data. Although several of these techniques may require a large amount of storage space (some require storing every data point), they may still be useful in some restricted sense, especially for judging the quality of simpler online models. One important nonparametric density estimation technique, kernel density estimation, is presented in Appendix A.

3.1 Recasting Previous Models in the Probabilistic Framework

Most of the local spatio–temporal background classifiers commonly used in video surveillance systems can be rewritten in terms of probability density functions. Below we describe how the three models described in [12] and presented in section 2.4 may be altered to represent background models as probability density functions.

3.1.1 Background of Constant Intensity

If the background distribution is assumed to have intensity I' at some pixel (x, y), we could potentially model the PDF of this background distribution as a single point mass at I'. Any measurement that deviated even slightly from I', however, would be scored lower than intended. A more useful approach would be to center a small Gaussian at I' that is scaled according to the noise present in the video.

Let $\hat{\sigma}_{GN}$ be an estimate of the Gaussian noise present in the training video data, and let

$$\mathbf{\Sigma} = \begin{pmatrix} \hat{\sigma}_{GN}^2 & 0 & 0 \\ 0 & \hat{\sigma}_{GN}^2 & 0 \\ 0 & 0 & \hat{\sigma}_{GN}^2 \end{pmatrix}$$

We then use the probability density function $f = N(I', \Sigma)$ to represent this background model.

There are several methods available for calculating a good estimate $\hat{\sigma}_{GN}$; one simple method is presented in section 3.2.1.

3.1.2 Background with Gaussian Distributions

The remaining two background models lend themselves readily to the probabilistic approach. Suppose that the background distribution at a pixel (x, y) is assumed to have a distribution given by a mixture of k Gaussians for some $k \in \mathbb{N}$. Let $\{\mu_i\}_{i=1}^k$ be the k mean values of the Gaussians, let $\{\mathbf{\Sigma}_i\}_{i=1}^k$ be the k covariance matrices of the Gaussians, and let $\{p_i\}_{i=1}^k$ be the k scaling factors associated with the mixture, with $0 \leq p_i \leq 1$ for $1 \leq i \leq k$ and $\sum_{i=1}^k p_i = 1$. Then we use the probability density function

$$f = \sum_{i=1}^{k} p_i \cdot N(\mu_i, \mathbf{\Sigma}_i)$$

to represent this background model. Notice that if $k = 1$ this degenerates to the single Gaussian model, $f = N(\mu, \mathbf{\Sigma})$.

3.2 Estimating the Probability that a Measurement Was Drawn from a Model Distribution

Suppose that at a particular pixel (x, y) we have generated a continuous probability density function $f(I_x, I_y, I_t)$ that approximates the true distribution of background measurements at (x, y). Given a new measurement $\mathbf{m} = (x, y, t)$, we wish to estimate the probability that $\mathbf{m}$ came from the distribution f. Simply evaluating $f(\mathbf{m})$ is meaningless, since f is a continuous probability density function. We can only make sense of average values of f on intervals.

Instead, we center a small Gaussian G at $\mathbf{m}$ and calculate the convolution $(G * f)(\mathbf{m})$ to estimate the probability that $\mathbf{m}$ was drawn from f.

It is not immediately clear how large the Gaussian G should be. One reasonable method would be to use estimates of the inherent noise in the (I_x, I_y, I_t) data as a guide. If $\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_t$ are estimates for the noise in the I_x, I_y, and I_t measurements, respectively, we can create the matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \hat{\sigma}_x^2 & 0 & 0 \\ 0 & \hat{\sigma}_y^2 & 0 \\ 0 & 0 & \hat{\sigma}_t^2 \end{pmatrix}$$

and set $G = N(\mathbf{m}, \mathbf{\Sigma})$ above. It may be impossible to estimate the noise inherent in the I_x, I_y, and I_t channels effectively. To estimate these values, however, we can use an estimate of the Gaussian noise present in the image space. There are several methods for doing so; one simple and fast method is presented below.

3.2.1 Estimating the Gaussian Noise in a Video

Immerkær [14] provides a fast method of estimating the standard deviation of additive Gaussian noise in an image, σ_{GN}. The approach is to convolve with a linear filter insensitive to the Laplacian of the image:

$$L = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

After applying L to an $m \times n$ image I, we may estimate σ_{GN} with:

$$\hat{\sigma}_{GN} = \sqrt{\frac{\pi}{2}} \frac{1}{6mn} \sum_{i,j=1}^{m,n} |(I * L)(i,j)|$$

To estimate the amount of noise present in a video sequence, we can simply measure $\hat{\sigma}_{GN}$ for a selection of frames and average the results.

3.3 Possible Problems with the Probabilistic Approach

The remainder of this paper deals with the problem of measuring the quality of an estimated background distribution function by comparing it with the true background distribution. Unfortunately, one cannot know the properties of the actual background distribution *a priori*. Moreover, it is incredibly difficult, if not impossible, to form a reasonable global distribution that is likely to model a wide range of natural scenes with any accuracy.

As a result, we need some way to approximate the true background distribution at a pixel. The problem of probability density estimation has received a great deal of attention over the past few decades, and many robust methods, both parametric and nonparametric, are available. As we cannot reasonably parameterize the background distributions encountered in natural scenes, we naturally choose a nonparametric approach.

The most popular nonparametric density estimation techniques include histograms, frequency polygons, average shifted histograms, and kernel density estimates. An excellent review of these methods may be found in [16]. Of these, kernel density estimation has the best efficiency; therefore, we use it exclusively for density estimation when needed. A review of multivariate kernel density estimation can be found in Appendix A.

Chapter 4

Quantitatively Measuring the Quality of Background Models

Given a background model of a scene, we wish to quantitatively measure how well the model reflects the actual background distribution. Suppose we have modeled the background of some video V and that for each pixel p the model is represented by a d-dimensional PDF, $\tilde{b}_p(\mathbf{x})$. Usually we select the $\tilde{b}_p$ models to lie within the three-dimensional (I_x, I_y, I_t) space, although different spaces could be used without altering the measures described below.

Whatever the underlying space, each pixel p is associated with an actual background distribution $b_p(\mathbf{x})$. In a good background model, the $\tilde{b}_p$ distributions will closely approximate the b_p distributions, especially in regions of the state space where measurements occur frequently. We describe a method to quantitatively measure this difference below. It is a a measure analogous to the relative entropy distance commonly used in information theory.

4.1 The Relative Entropy Measure

The relative entropy measure is widely used in information theory, where it has a very specific meaning. Suppose that we wish to encode the values of a random variable

with distribution f. If we could completely specify the distribution f, we could encode the random variable with a code that has average length $H(f)$ bits, where $H(f)$ is the Shannon entropy of f [1]. If, however, f were not completely specifiable, we could use an approximate distribution g to model the random variable. In that case, the code would need more bits to represent the random variable. That difference is exactly the entropy of g relative to f, $d(f||g)$.

The relative entropy measure, by definition, calculates the information lost when we approximate one probability density function with another. This is the reason we choose it as the basis of our quality measure.

4.1.1 Definition of Relative Entropy

Suppose Ω is a measurable state space and that f and g are probability densities on Ω. The entropy of g relative to f, also called the Kullback-Leibler distance, is given by

$$d(f||g) = \int_{S(f)} f(\omega) \log \left(\frac{f(\omega)}{g(\omega)} \right) d\omega$$

where $S(f)$ is the support of f [3]. The common convention, reached by appealing to continuity arguments, is to define $f \cdot \log\left(\frac{f}{0}\right) = \infty$ for nonzero f [3]. The relative entropy measure, then, assumes values in $[0, \infty]$.

The relative entropy measure is not a true metric. It is not symmetric and does not satisfy the triangle inequality. Nevertheless, it has many useful properties. It is always nonnegative and only assumes the value 0 if $f = g$ almost everywhere on Ω [1]. Since this property is usually only proven for the univariate case (with $\Omega = \mathbb{R}$), we prove this property below.

Theorem: Let f and g be probability density functions on the measurable space Ω. Then $d(f||g) \geq 0$, with equality holding if and only if $f = g$ almost everywhere on Ω.

Proof: We have:

$$\begin{aligned} -d(f||g) &= -\int_{S(f)} f(\omega) \log\left(\frac{f(\omega)}{g(\omega)}\right) d\omega \\ &= \int_{S(f)} f(\omega) \log\left(\frac{g(\omega)}{f(\omega)}\right) d\omega \\ &\leq \log \int_{S(f)} f(\omega) \left(\frac{g(\omega)}{f(\omega)}\right) d\omega \\ &= \log \int_{S(f)} g(\omega) d\omega \\ &\leq \log 1 \\ &= 0 \end{aligned}$$

The first inequality follows from Jensen's inequality [19]. If equality is to hold there, we must have that the argument of the convex function $\log(\cdot)$ be a constant almost surely, that is, that $f = g$ almost everywhere. □

4.2 A New Measure of Background Model Quality

We modify the relative entropy measure slightly to measure the distance between the $\tilde{b}_p$ and b_p distributions. Unfortunately, we cannot know *a priori* how the background measurements are distributed, and so cannot calculate directly with the b_p distributions. Instead we choose from a variety of nonparametric density estimation techniques to form good approximations of the b_p distributions.

Our modification will reflect the heuristic observation made above, that is, that we should penalize background models heavily for differing from the true distribution in areas where measurements occur relatively frequently, but should not penalize too harshly for differing from the true distribution in areas where measurements occur relatively infrequently. To do so, for each pixel p we build an approximation of the universal distribution of measurements at p, including both background and foreground measurements. For a particular pixel p, label this universal distribution $f_p(\mathbf{x})$.

4.2.1 Definition of Relevance-Weighted Relative Entropy

Choose and fix a pixel $p = (x, y)$ in the image. Using nonparametric density estimation, we build approximations of $f_p(\mathbf{x})$ and $b_p(\mathbf{x})$. Call these approximations $\hat{f}(\mathbf{x})$ and $\hat{b}(\mathbf{x})$, respectively. Once suitable $\hat{f}$ and $\hat{b}$ have been found, we calculate the *relevance-weighted relative entropy* of the model $\tilde{b}_p$ as:

$$q(\tilde{b}_p) = \int_X \hat{f}_p(x) \log \left(\frac{\hat{b}_p(\mathbf{x})}{\tilde{b}_p(\mathbf{x})} \right) dx \tag{4.1}$$

The measure $q(\tilde{b}_p)$ is based on the relative entropy measure between $\tilde{b}_p$ and $\hat{b}_p$, modified to weight their difference by the function $\hat{f}$.

It is easy to see that although this measure is not a true metric, it equals 0 if $\tilde{b}_p = \hat{b}_p$.

In the next chapter we present results from natural scenes that suggest that relative quality is a useful measure to consider.

Chapter 5

Results

The new quality measure was extensively tested with several parametric and non-parametric density estimation techniques. The relative performance of each of the methods, as measured by their relevance-weighted relative entropies, closely matches their expected performance. This suggests that the new measure is likely to be useful as an unbiased estimator of a background model's quality.

We briefly describe the test sequences used and background models tested in the next two sections.

5.1 Description of Test Sequences

We chose three test videos to represent several common scenarios where background subtraction might be used. For each sequence, we concentrate on specific image regions that were chosen to reflect differing amounts and types of activity. We list the videos and brief descriptions below.

- Ducks—The first video is a scene of ducks swimming on a pond. The camera remains static, but both the water and surrounding foliage are affected by blowing wind. This is an example of a video that contains a great deal of fairly small, sometimes inconsistent motion (swaying grasses) that should be ignored when

searching for the true anomalies (the ducks). Although the ducks are benign, this type of scene typifies many surveillance situations.

The first region of interest lies in a portion of the water where ducks occasionally wander; the second region focuses entirely on waving grass.

- Intersection—The second video is a scene of a traffic intersection. A great deal of relatively consistent motion occurs, except in the intersection's center. This is an example of a possible commercial use for automated surveillance—analyzing traffic flow patterns.

  The first region of interest is in the center of the intersection where a lot of activity occurs in different directions of motion; the second region lies in a region of sidewalk with little activity. We show a sample frames from this videos, highlighting the region(s) of interest, in Figure 1.

- Stabilized—The third video is a stabilized aerial view of a suburban area. This is an example another possible use of automated surveillance systems—modeling consistent motion in large, mostly static scenes.

  The region of interest is for the most part static, but a small portion contains a road on which cars consistently drive.

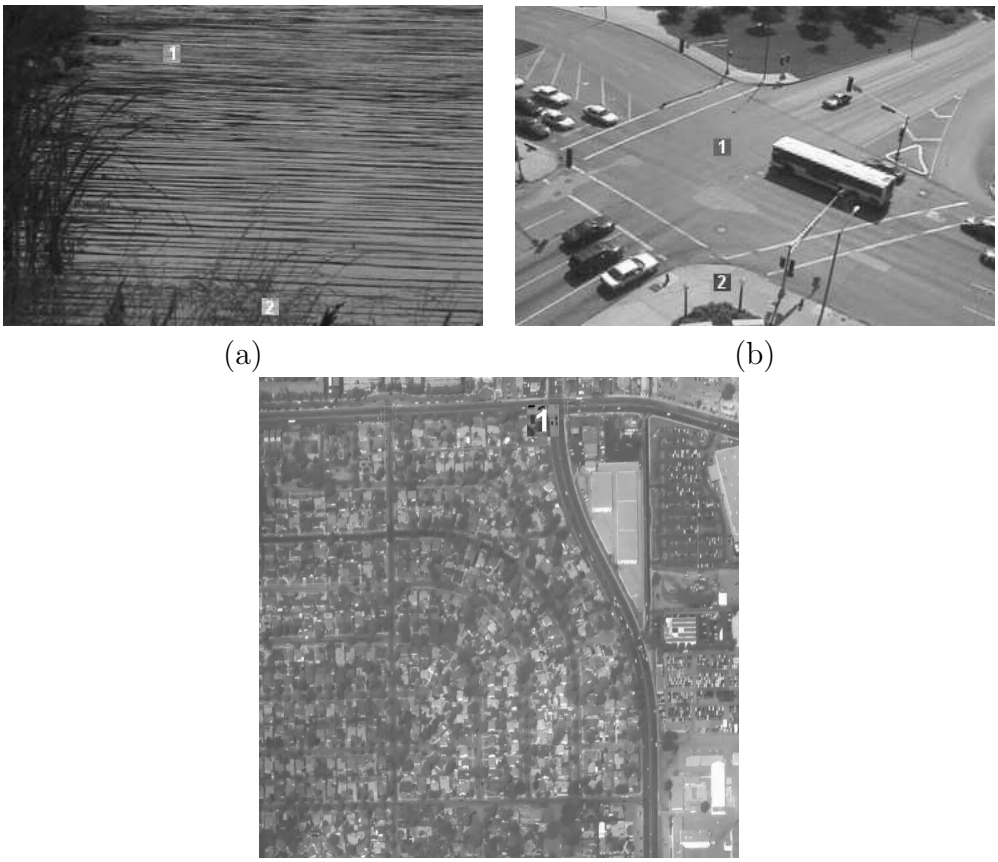
5.2 Description of Background Models Tested

The relevance-weighted relative entropy measure was tested on seven local spatio–temporal background estimation techniques. We list them, including any relevant information, below.

Parametric models:

- Uniform intensity

- Single Gaussian in (I_x, I_y, I_t) space



(a) (b)

(c)

Figure 5.1: Frames from the test sequences: (a) "ducks," (b) "intersection," and (c) "stabilized." Regions of interest are inverted and labeled.

- Mixture of 3 Gaussians in (I_x, I_y, I_t) space (generated with expectation maximization algorithm)
- Mixture of up to 5 Gaussians in (I_x, I_y, I_t) space (generated with adaptive mixtures technique)

Nonparametric models:

- Kernel density estimate with 10 data points randomly chosen from training data
- Same with 30 points
- Same with 50 points

5.3 Procedure

After selecting a portion of the video sequence to represent the background, we used that portion to train and construct probability density estimates for each pixel of interest, using the methods described previously and in the appendix. These estimates comprised the $\tilde{b}$ distributions in the definition of relevance-weighted relative entropy. Additionally, for each pixel one large universal model was created, using kernel density estimation, with the data from the entire training sequence. This large background distribution was used to represent the $\hat{f}$ distributions. A subsample of the training sequence was used to create a smaller model, again using kernel density estimation, to represent the $\hat{b}$ distributions in the definition.

With these distributions, we proceeded to calculate the relevance-weighted relative entropies for each of the background models as given in equation 4.1. The results for each video region and background model are presented in Table 1. The number listed is the mean relevance-weighted relative entropy of the model over the entire region being tested.

To evaluate the integral in equation 4.1, Monte Carlo integration was used with enough sample points to ensure that the result was very likely to be accurate to within 5% error.

Table 5.1: Relevance-weighted relative entropy measures for several density estimates across differing video sequences

Method	Ducks		Intersection		Stabilized
	Region 1	Region 2	Region 1	Region 2	Region 1
Uniform	1.758×10^{-3}	1.220×10^{-3}	8.262	2.245	0.2207
Single Gaussian	2.036×10^{-4}	1.681×10^{-3}	9.634	4.030	0.2917
Multiple Gaussian (AM)	4.734×10^{-2}	4.161×10^{-2}	5.959	3.703	0.5056
Multiple Gaussian (EM)	1.479×10^{-4}	1.567×10^{-3}	4.365	3.228	0.2955
KDE (10 points)	0.6820	0.7262	1.0404	0.7377	0.5068
KDE (30 points)	0.3722	0.3445	0.3877	0.3968	0.1987
KDE (50 points)	0.2470	0.2582	0.3300	0.3311	0.1234

5.4 Findings

Of the parameterized methods, the multiple Gaussian mixture model built with expectation maximization was the best in regions with a lot of activity. For regions with little activity, however, simpler methods such as single Gaussian models or even uniform intensity models performed slightly better. The multiple Gaussian mixture model built with adaptive mixtures performed similarly, suggesting that it may be useful as an online substitute for expectation maximization.

The single Gaussian and uniform models performed surprisingly well. This should not be that surprising, however, since in many of these background distributions outliers are fairly scarce.

Additionally, the relevance-weighted relative entropy measures are greater for regions that have more activity, which agrees with the common sense notion that distributions with larger support are more difficult to model with highly localized representations.

Somewhat surprisingly, the nonparametric density estimators produced very good results. For the "intersection" scene, for example, the kernel density estimates performed by far the best, even with a very small amount of data. These models can, however, be understood as being nothing more than large mixture models that

use identical scaling for each mixture. In fact, if infinite support is desirable, the Epanechnikov kernels we use could be replaced with standard normal distributions.

It should also be noted that the relevance-weighted relative entropies did not vary greatly over the test region. Any variation more than slight could be resolved easily—higher entropies generally corresponded to more active pixels.

These results suggest that kernel density estimators or other nonparametric estimates of density should be strongly considered when deciding upon a background model. They model a wide range of datasets quite well and require little storage space with small datasets. A kernel density estimate with 10 points, for example, only requires storing $10 \cdot d$ floating point numbers, where d is the dimension of the space. A finite mixture model with 5 Gaussians, for comparison, requires storing $5d^2 + 6d$ floating point numbers.

5.5 Implementation

All code was written in MATLAB. To create kernel density estimates, the Kernel Density Estimation Toolbox for MATLAB, created by Alexander Ihler, was used [7]. Additionally, the expectation maximization and adaptive mixtures techniques were implemented using the Computational Statistics Toolbox, created to accompany a textbook by Wendy L. Martinez and Angel R. Martinez, was used, despite several bugs in their code [8].

Chapter 6

Conclusion

Although there are several classifier-based methods that can be used successfully for background subtraction, they are all lacking a theoretical basis. Additionally, classifier-based techniques have several weaknesses that should not be ignored. One particularly important weakness is that no good method exists to quantitatively measure the quality of a given background model.

After adopting a framework based on probability density functions, we developed a technique for measuring the relevance-weighted relative entropy of a background model. The technique is based on an important information-theoretic quantity, the relative entropy.

Using kernel density estimation to model the true background and universal distributions at a pixel, we tested the newly developed relevance-weighted relative entropy measure on several parametric and nonparametric density estimation methods. Even for very different video regions, the relevance-weighted relative entropy measure supports the traditionally-held ranking of the qualities of these techniques, supporting the notion that the new measure is indeed useful.

Appendix A

Kernel Density Estimation

Popular nonparametric density estimation schemes include histograms, frequency polygons, average shifted histograms, and kernel density estimation. Of these, kernel density estimation is the most efficient, converging to the true distribution (in terms of asymptotic mean integrated squared error) at a rate of $O(n^{-\frac{4}{4+d}})$, where n is the number of data points used to build the estimate and d is the dimension of the state space. The multivariate histogram, for comparison, converges at a rate of $O(n^{-\frac{2}{4+d}})$. The other methods mentioned above converge at the same rate as kernel density estimates, but with larger constants [16].

Somewhat surprisingly, kernel density estimates, may be understood as the limiting case of these other methods. Frequency polygons and average shifted histograms asympototically approximate kernel density estimators as their bin width becomes smaller. In fact, almost every density estimation technique asymptotically becomes a kernel estimate [16].

For our purposes, one of the most useful properties of a kernel density estimate is that it gives a bona fide estimate of the true probability density—the estimate is always guaranteed to be a proper probability density itself.

The idea is to center a small, scaled kernel at each data point. To estimate the probability density function at a new measurement, we evaluate the kernels at that point and sum the results.

A.1 Definition and Discussion of Parameter Selection

A.1.1 Definition

Choose and fix a pixel (x, y) and select an n-frame section of the training data from which to build an approximate background distribution. Decide upon a particular state space for the distributions to lie within, and let $\{\mathbf{m}_i\}_{i=1}^{n}$ be the measurements in that space at the pixel (x, y) for each of the n frames. If f is the true background distribution in this space at the pixel (x, y), we can estimate the value of f at some point $\mathbf{x}$ in the domain by calculating as follows:

$$\hat{f}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\prod_{j=1}^{d} h_j^{-1} K\left(\frac{\mathbf{x}_j - \mathbf{m}_{i,j}}{h_j}\right)\right\}$$

The values $\{h_j\}_{j=1}^{d}$ are called the scaling factors of the kernel estimate, and the function $K(t)$ is a one-dimensional kernel function that satisfies the properties listed below:

- $K(t) > 0$ for all $t \in \mathbb{R}$
- $K(t) = K(-t)$ for all $t \in \mathbb{R}$
- $\int_{-\infty}^{\infty} K(t)dt = 1$.

A.1.2 Choice of Kernel Function

The paricular choice of kernel is not terribly important in terms of efficiency and convergence [16]. Nonetheless, other factors, including computational cost, should be carefully considered when deciding between available kernels. Epanechnikov proved that the most efficient univariate kernel is given by his eponymous kernel, $K(t) = \frac{3}{4}(1-t^2)\cdot I_{[-1,1]}(t)$, where $I_{[-1,1]}$ is an indicator function [2]. We select the Epanechnikov kernel for our kernel because it is optimally efficient, is simple to calculate, and has

compact support—to calculate $\hat{f}(\mathbf{x})$ we only need to consider measurements within L^1 distance $\max_j\{h_j\}$ from $\mathbf{x}$.

A.1.3 Choice of Scaling Parameters

The quality of the kernel estimate $\hat{f}$ strongly depends on the choice of scaling factors. Scott shows how to optimally estimate the scaling parameters $\{h_j\}$ [16]. Suppose K is a kernel with finite and nonzero second central moment κ and f is the density function we wish to approximate. The idea is to minimize the asymptotic mean integrated squared error (AMISE) of the kernel estimate $\hat{f}$ given a particular set of scaling factors $\{h_j\}$. For the univariate case, he calculates

$$AMISE_K(h) = \frac{1}{4}\kappa^2 h^4 R(f'') + \frac{R(K)}{nh}$$

The function R is a "roughness" measure, defined for a function f as

$$R(f) \equiv \int_{-\infty}^{\infty} f(x)^2 dx$$

Minimizing the AMISE function yields the optimal scaling factor:

$$h_K^* = \left(\frac{R(K)}{n\kappa^2 R(f'')}\right)^{\frac{1}{5}}$$

Since f is in general unknown, we must approximate the value of $R(f'')$. The usual method (given by Silverman [17]) is to assume f has the standard normal distribution $N(0, \sigma_f^2)$, where σ_f is the standard deviation of f. This leads to the estimate $R(f'') \approx 3/(8\sqrt{\pi})\sigma_f^5$. We can approximate σ_f using the sample standard deviation of the data or the nonparametric estimator $IQR/1.348$; Silverman recommends using the smaller of the two [17].

For the Epanechnikov kernel K_E, we have $\kappa_E = 1/5$ and $R(K_E) = 3/5$, leading to the estimate

$$h^*_{K_E} = (40\sqrt{\pi})^{\frac{1}{5}}\sigma \approx 2.34\sigma n^{\frac{1}{5}}$$

Unfortunately, the optimal scaling parameters cannot be directly calculated for multivariate kernels. Instead, Scott explicitly calculates the optimal scaling parameters $\{h_j\}$ for the Normal kernel $N(0, I_d)$, where I_d is the $d \times d$ identity matrix. He derives the Normal reference rule:

$$h^*_j = \left(\frac{4}{d+2}\right)^{frac1d+4} \sigma_j n^{-\frac{1}{d+4}}$$

Scott then suggests that for other kernels, the scaling parameters are given by:

$$\hat{h}^*_j(K) = \frac{h^*_j}{\sigma_K}$$

Therefore, for the Epanechnikov kernel K_E, we may use

$$\hat{h}^*_j(K_E) = \sqrt{5} \cdot h^*_j$$

as our approximately optimal scaling factors.

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[2] V. K. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Applied Probability*, 14:153–158.

[3] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International Statistics Review*, 70(3):419–436.

[4] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 22, 1998.

[5] I. Haritaoglu, D. Harwood, and L. Davis. A real-time system for detecting and tracking people in 2.5d. In *Proceedings of the European Conference on Computer Vision*, 1998.

[6] T. Horprasert, D. Harwood, and L. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV FRAME-RATE Workshop*.

[7] A. Ihler. Kernel density estimation toolbox for MATLAB. Available online: http://ssg.mit.edu/~ihler/code/kde.shtml/.

[8] W. L. Martinez and A. R. Martinez. *Computational Statistics Handbook with MATLAB*. Chapman & Hall / CRC Press, 2002.

[9] T. Matsuyama, T. Ohya, and H. Habe. Background subtraction for non-stationary scenes. In *Proceedings of the 4th Asian Conference on Computer Vision*, pages 662–667, 2000.

[10] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. In *Proceedings of the International Conference on Vision Systems*, 1999.

[11] R. Pless, T. Brodsky, and Y. Aloimonos. Detecting independent motion: The statistics of temporal continuity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):68–73, 2000.

[12] R. Pless, J. Larson, S. Siebers, and B. Westover. Evaluation of local models of dynamic backgrounds. In *Proceedings of the IEEE Conference on Computer Vision and Patter Recognition*, 2003.

[13] C.E. Priebe. Adaptive mixture density estimation. *Journal of the American Statistical Association*, 89:796–806.

[14] J. Immerkær. Fast noise variance estimation. *Computer Vision and Image Understanding*, 64:300–302, 1996.

[15] A. R. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26:195–239.

[16] D. W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. John Wiley & Sons, Inc., 1992.

[17] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

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

[19] R. L. Wheeden and A. Zygmund. *Measure and Integral*. Number 43 in Monographs and Textbooks in Pure and Applied Mathematics. Dekker, 1977.

[20] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):774–780, 2000.

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

Vita

Roman Garnett

Date of Birth May 20, 1982

Place of Birth Long Beach, CA

Awards and Honors
Sigma Xi
Tau Beta Pi
Department Chairman's Award (Computer Science)
Antionette Frances Dames Award (School of Engineering)
Ross Middlemiss Prize (Mathematics)
Deparmental Honors (Mathematics)

Publications
R. Garnett, T. Huegerich, C. Chui, and W. He. "A New Framework for the Removal of Gaussian and Impulse Noise." Submitted to *IEEE Transactions on Image Processing*, March 2003. Preprint available.

May 2004