EURASIP Journal on Applied Signal Processing 2005:14, 2281–2291

Spatio-temporal Background Models for Outdoor Surveillance

Robert Pless

Department of Computer Science and Engineering, Washington University in St. Louis, MO 63130, USA
Email: pless@cse.wustl.edu

Received 2 January 2004; Revised 1 September 2004

Video surveillance in outdoor areas is hampered by consistent background motion which defeats systems that use motion to identify intruders. While algorithms exist for masking out regions with motion, a better approach is to develop a statistical model of the typical dynamic video appearance. This allows the detection of potential intruders even in front of trees and grass waving in the wind, waves across a lake, or cars moving past. In this paper we present a general framework for the identification of anomalies in video, and a comparison of statistical models that characterize the local video dynamics at each pixel neighborhood. A real-time implementation of these algorithms runs on an 800 MHz laptop, and we present qualitative results in many application domains.

Keywords and phrases: anomaly detection, dynamic backgrounds, spatio-temporal image processing, background subtraction, real-time application.

1. INTRODUCTION

Computer vision has had the most success in well-constrained environments. Well constrained environments allow the use of significant prior expectations, explicit or controlled background models, easily detectable features, and effective closed-world assumptions. In many surveillance applications, the environment cannot be explicitly controlled and may contain significant and irregular motion. However irregular, the natural appearance of a scene as viewed by a static video camera is often highly constrained. Developing representations of these constraints—models of the typical (dynamic) appearance of the scene—will allow significant benefits to many vision algorithms. These models capture the dynamics of video captured from a static camera of scenes such as trees waving in the wind, traffic patterns in an intersection, and waves over water. This paper develops a framework for statistical models to represent dynamic scenes.

The approach is based upon spatio-temporal image analysis. This approach explicitly avoids finding or tracking image features. Instead, the video is considered to be a 3D function giving the image intensity as it varies in space (across the image) and time. The fundamental atoms of the image processing are the value of this function and the response to spatio temporal filters (such as derivative filters), measured at each pixel in each frame. Unlike interest points or features, these measurements are defined at every pixel in the video sequence. Appropriately designed filters may give robust measurements to form a basis for further processing. Optimality criteria and algorithms for creating derivative and blurring filters of a particular size and orientation lead to significantly better results than estimating derivatives by applying Sobel filters to raw images [1]. For these reasons, spatio-temporal image processing is an ideal first step for streaming video processing applications.

Calculating (one or more) filter responses centered at each pixel in a video sequence gives a representation of the appearance of the video. If these filters have a temporal component (such as a temporal derivative filter), then the joint distribution of the filter responses can model dynamic features of the local appearance of the video. Maintaining the joint distribution of the filter responses gives a statistical model for the appearance of the video scene. When the same filters are applied to new video data, a score is computed that indicates how well they fit the statistical appearance model. This is our approach to finding anomalous behavior in a scene with significant background motion.

Four facts make this approach possible. First, appropriate representations of the statistics of the video sequence can give quite specific characterizations of the background scene. This allows the theoretical ability to detect a very large class of anomalous behavior. Second, these statistical models can be evaluated in real time on nonspecialized computing hardware to make an effective anomaly detection system. Third, effective representations of very complicated scenes can be maintained with minimal memory requirements—linear in the size of the image, but independent of the length of the video used to define the background model. Fourth, for an arbitrary video stream, the representation can be generated and updated in real time, allowing the model the freedom (if desired) to adapt to slowly varying background conditions.

1.1. Streaming video

The emphasis in this paper is on streaming-video algorithms—autonomous algorithms that run continuously for very long time periods that are real time and robust. Streaming-video algorithms have specific properties and constraints that help characterize their performance, including (a) the maximum memory required to store the internal state, (b) per-frame computation time that is bounded by the frame-rate, and, commonly (c) an output structure that is also streaming, although it may be either a stream of images or symbols describing specific features of the image. These constraints make the direct comparison of streaming algorithms to offline image analysis algorithms difficult.

1.2. Roadmap to paper

Section 2 gives a very brief overview of other representative algorithms. Section 4 presents our general statistical approach to spatio-temporal anomaly detection, and Section 5 gives the specific implementation details for the filter sets and nonparametric probability density representations that have been implemented in our real-time system. Qualitative results of this real-time algorithm are presented for a number of different application domains, and quantitative results in terms of ROC plots for the domain of traffic pattern analysis.

2. PRIOR WORK

The framework of many surveillance systems is shown in Figure 1. This work is concerned with the development and analysis of the background model. Each background model defines an error measure that indicates if a pixel is likely to come from the background. The analysis of new video data consists of calculating this error for each pixel in each frame. This measure of error is either thresholded to mark objects that do not fit the background model, enhanced with spatial or temporal integration, or used in higher-level tracking algorithms. An excellent overview and integration of different methods for background subtraction can be found in [2].

Surveillance systems generate a model of the background and subsequently determine which parts of (each frame of) new video sequences fit that model. The form of the background model influences the complexity of this problem, and can be based upon (a) the expected color of a pixel [3, 4] (e.g., the use of blue screens in the entertainment industry), or (b) consistent motions, where the image is static [5] or undergoing a global transformation which can be affine [6] or planar projective [7]. Several approaches exploit spatio-temporal intensity variation for more specific tasks than general anomaly detection [8, 9]. For the specific case of gait recognition, searching for periodicity in the spatio-temporal intensity signal has been used to search for people by detecting gait patterns [10].

This paper most explicitly considers the problem of developing background models for scenes with consistent background motion. A very recent paper [11] considers the same question but builds a different kind of background model. These background models are global models of image variation based on dynamic textures [12]. Dynamic textures represent each image of a video as a linear combination of basis images. The parameters for each image define a point in a parameter space, and an autoregressive moving average is used to predict parameters (and therefore the appearance) of subsequent frames. Pixels which are dissimilar from the prediction are marked as independent and tracked with a Kalman filter. Our paper proposes a starkly different background model that models the spatio-temporal variance locally at each pixel. For dynamic scenes, such as several trees waving independently in the wind, water waves moving across the field of view, or complicated traffic patterns, there is no small set of basis images that accurately captures the degrees of freedom in the scene. For these scenes, a background model based on global dynamic textures will either provide a weak classification system or require many basis images (and therefore a large state space).

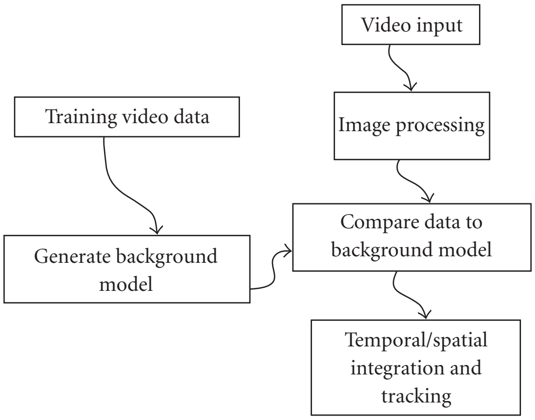
Video input
Training video data
Image processing
Generate background model
Compare data to background model
Temporal/spatial integration and tracking

Figure 1: The generic framework of the front end of visual surveillance systems. This work focuses on exploring different local background models.

Finally, qualitative analyses of local image changes have been carried out using oriented energy measurements [13]. Here we look at the quantitative predictions that are possible with similar representations of image variation. This paper does not develop or present a complete surveillance system. Rather, it explores the statistical and empirical efficacy of a collection of different background models. Each background model produces a score for each pixel that indicates the likelihood that the pixel comes from the background. Classical algorithms that use the difference between a current pixel and a background image pixel as a first step can simply incorporate this new background model and become robust to consistent motions in the scene.

3. A REPRESENTATION OF DYNAMIC VIDEO

In this section we present a very generic approach to anomaly detection in the context of streaming video analysis. The concrete goal of this approach has two components. First, for an input video stream, develop a statistical model of the appearance of that stream. Second, for new data from the same stream, define a likelihood (or, if possible, a probability) that each pixel arises from the appearance model. We assume that the model is trying to represent the "background" motion of the scene, so we call the appearance model a background model.

In order to introduce this approach, we start with several definitions which make the presentation more concrete. The input video is considered to be a function I, whose value is defined for different pixel locations (x, y), and different times t. The pixel intensity value at pixel (x, y) during frame t will be denoted by $I(x, y, t)$. This function is a discrete function, and all image processing is done and described here in a discrete framework. However, the justification for using discrete approximations to derivative filters is based on the view of I as a continuous function.

A general form of initial video processing is computing the responses of filters at all locations in the video. The filters we use are defined as an $n \times n \times m$ array, and $F(i, j, k)$ denotes the value of the (i, j, k) location in the array. For simplicity, we assume that n is odd. The response to a filter F will be denoted by I_F, and the pixel location x, y, t of I_F is defined to be

$$\begin{aligned} I_F(x, y, t) &= \sum_{i=1,\dots,n} \sum_{j=1,\dots,n} \sum_{k=1,\dots,m} I\Big(x + i - \frac{n-1}{2}, y \\ &\qquad + j - \frac{n-1}{2}, t - k + 1\Big) F(i, j, k). \end{aligned} \tag{1}$$

This filter response is centered around the pixel (x, y), but has the time component equal to the latest image used in computing the filter response. Defining a number of spatio-temporal filters and computing the filter response at each pixel in the image captures properties of the image variation at each pixel. Which properties are captured depends upon which filters are used—the next section picks a small number of filters and justifies why they are most appropriate for some surveillance applications. However, a general approach to detecting anomalies at a specific pixel location (x, y) may proceed as follows:

(i) define a set of spatio-temporal filters $\{F_1, F_2, \dots, F_s\}$;

(ii) during training, capture the vector of measurements $\vec{m}_t$ at each frame t as $\langle F_1(x, y, t), F_2(x, y, t), \dots, F_s(x, y, t)\rangle$. The first several frames will have invalid data until there are enough frames so that the spatio-temporal filter with greatest temporal extent can be computed. Similarly, we ignore edge effects for pixels that are close enough to the boundary so that the filters cannot be accurately computed;

(iii) individually for each pixel, consider the set of measurements for all frames in the training data $\{\vec{m}_1, \vec{m}_2, \dots\}$ to be samples from some probability distribution. Define a probability density function P on the measurement vector so that $P(\vec{m})$ gives the probability that measurement $\vec{m}$ comes from the background model.

We make this abstract model more concrete in the following section; however, this model encodes several explicit design choices. First, all the temporal variation in the system is captured explicitly in the spatio-temporal filters that are chosen. It is assumed that the variation in the background scene is independent of the time, although in practice the probability density function can be updated to account for slow changes to the background distribution. Second, the model is defined completely independently for each pixel and therefore may give very accurate delineations of where behavior is independent. Third, it outputs probabilities or likelihoods that a pixel is independent, exactly like prior background subtraction methods, and so can be directly incorporated into existing systems.

4. MODELS OF BACKGROUND MOTION

For simplicity of notation, we drop the (x, y) indices, but we emphasize that background model presented in the following section is independently defined for each pixel location. The filters chosen in this case are spatio-temporal derivative filters. The images are first blurred with a 5-tap discrete Gaussian filter with standard deviation 1.5. Then we use the optimal 7-tap directional derivative filters as defined in [1] to compute the spatial derivatives I_x, I_y, and frame-to-frame differencing of consecutive (blurred) images to compute the temporal derivative I_t. Thus every pixel in every frame has an image measurement vector of the form $\langle I, I_x, I_y, I_t\rangle$, the blurred image intensity, and the three derivative estimates, computed by applying the directional derivative filters to this blurred image.

This filter set is chosen to be likely to contain much of the image variation because it is the zeroth- and first-order expansion of the image intensity around each pixel. Also, one mode of common image variation is consistent velocity motion at a given pixel. In this case, regardless of the texture of an object moving in a particular direction, the $\langle I_x, I_y, I_t\rangle$ components lie on a plane in the spatio-temporal derivative space (which plane they lie on is dependent upon the velocity). Representing this joint distribution accurately means that any measured spatio-temporal derivative that is significantly off this plane can be marked as independent. That is, we can capture, represent, and classify a motion vector at a particular pixel without ever explicitly computing optic flow. Using this filter set, the following section defines a number of different methods for representing and updating the measurement vector distribution.

Each local model of image variation is defined with four parts: first, the measurement—which part of the local spatio-temporal image derivatives the model uses as input; second, the score function which reports how well a particular measurement fits the background model; third, the estimation procedure that fits parameters of the score function to a set of data that is known to come from the background; fourth, if applicable, an online method for estimating the parameters of the background model, so that the parameters can be updated for each new frame of data within the context of streaming video applications.

4.1. Known intensity

The simplest background model is a known background. This occurs often in the entertainment or broadcast television industry in which the environment can be engineered to simplify background subtraction algorithms. This includes the use of "blue screens," backdrops with a constant color which are designed to be easy to segment.

Measurement

The measurement $\vec{m}$ is the color of a given pixel. For the gray-scale intensity, the measurement consists just of the intensity value: $\vec{m} = I$. For color images, the value of m is the vector of the color components $\langle r, g, b\rangle$, or the vector describing the color in the HSV or another color space.

Score

Assuming Gaussian zero-mean noise with variance σ^2 in the measurement of the image intensity, the negative log-likelihood that a given measurement $\vec{m}$ arises from the background model is $f(\vec{m}) = (\vec{m} - \vec{m}_{\text{background}})^2/\sigma^2$. The score function for many of the subsequent models has a probabilistic interpretation, given the assumption of Gaussian noise corrupting the measurements. However, since the assumption of Gaussian noise is often inaccurate and since the score function is often simply thresholded to yield a classification, we do not emphasize this interpretation.

Estimation

The background model $\vec{m}_{\text{background}}$ is assumed to be known a priori.

4.2. Constant intensity

A common background model for surveillance applications is that the background intensity is constant, but initially unknown.

Measurement

The gray-level intensity (or color) of a pixel in the current frame is the measurement $\vec{m} = I$ or $\vec{m} = \langle r, g, b\rangle$.

Score

The independence score for this model is calculated as the Euclidean distance of the measurements from the mean $f(\vec{m}) = ||\vec{m} - \vec{m}_\mu||_2^2$.

Parameter estimation

The only parameter is the estimate of the background intensity. m_μ is estimated as the average of the measurements taken of the background.

Online parameter estimation

An online estimation process maintains a count n of the number of background frames and the current estimate of m_μ. This estimate can be updated: $\vec{m}_{\mu_{\text{new}}} = ((n-1)/n)\vec{m}_\mu + (1/n)\vec{m}$.

4.3. Constant intensity and variance

If the background is not actually constant, then modeling both the mean intensity at a pixel and its variance gives an adaptive tolerance for some variation in the background.

Measurement

The gray-level intensity (or color) of a pixel in the current frame is the measurement $\vec{m} = I$ or $\vec{m} = \langle r, g, b\rangle$.

Model parameters

The model parameters consist of the mean measurement $\vec{m}_\mu$ and the variance σ^2.

Score

Assuming Gaussian zero-mean noise with variance σ in the measurement of the image intensity, the negative log-likelihood that a given measurement $\vec{m}$ arises from the background model is $f(\vec{m}) = ||\vec{m} - \vec{m}_\mu||_2^2/\sigma^2$.

Parameter estimation

For the given set of background samples, the mean intensity $\vec{m}_\mu$ and the variance σ^2 are computed as the average and variance of the background measurements.

Online parameter estimation

The online parameter estimation for each of the models can be expressed in terms of a Kalman filter. However, since we have the same confidence in each measurement of the background data, it is straightforward and instructive to write out the update rules more explicitly. In this case, we maintain a count n, the current number of measurements. The mean $\vec{m}_\mu$ is updated so that $\vec{m}_{\mu_{\text{new}}} = (1/(n+1))\vec{m} + (n/(n+1))\vec{m}_\mu$. If each measurement is assumed to have variance 1, the variance σ^2 is updated as follows: $\sigma^2_{\text{new}} = (1/\sigma^2 + 1)^{-1}$.

4.4. Gaussian distribution in $\langle I, I_x, I_y, I_t\rangle$-space

The remainder of the models use the intensity and the spatio-temporal derivatives of intensity in order to make a more specific model of the background. The first model of this type uses a Gaussian model of the distribution of measurements in this space.

Measurement

The 4-vector consisting of the intensity and the x, y, t derivatives of the intensity is $\vec{m} = \langle I, I_x, I_y, I_t\rangle$.

Model parameters

The model parameters consist of the mean measurement $\vec{m}_\mu$ and the covariance matrix Σ.

Score

The score for a given measurement $\vec{m}$ is

$$f(\vec{m}) = (\vec{m} - \vec{m}_\mu)^{\mathrm{T}}\Sigma^{-1}(\vec{m} - \vec{m}_\mu). \tag{2}$$

Estimation

For a set of background measurements $m_1, \dots, m_k$, the model parameters can be calculated as

$$\vec{m}_\mu = \frac{\sum_{i=1,\dots,k} m_i}{k}, \qquad \Sigma = \frac{\sum_{i=1,\dots,k} (m_i - \vec{m}_\mu)(m_i - \vec{m}_\mu)^{\mathrm{T}}}{k-1}. \tag{3}$$

Online estimation

The mean value $\vec{m}_\mu$ can be updated by maintaining a count of the number of measurements so far as in the previous model. The covariance matrix can be updated incrementally:

$$\Sigma_{\text{new}} = \frac{n}{n+1}\Sigma + \frac{n}{(n+1)^2}(\vec{m} - \vec{m}_\mu)(\vec{m} - \vec{m}_\mu)^{\mathrm{T}}. \tag{4}$$

4.5. Multiple Gaussian distribution in $\langle I, I_x, I_y, I_t \rangle$-space

Using several multidimensional Gaussian distributions allows a greater freedom to represent the distribution of measurements occurring in the background. An EM algorithm is used to fit several (the results in Section 5 use three) multidimensional Gaussian distributions to the measurements at a particular pixel location [14, 15].

Model parameters

The model parameters are the mean value and covariance for a collection of Gaussian distributions.

Score

The score for a given measurement $\vec{m}$ is the distance from the closest of the distributions:

$$f(\vec{m}) = \min_i (\vec{m} - \vec{m}_{\mu_i})^{\mathrm{T}} \Sigma_i^{-1} (\vec{m} - \vec{m}_{\mu_i}). \tag{5}$$

Online estimation

We include this model because its performance was often the best among the algorithms considered. To our knowledge, however, 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.

4.6. Constant optic flow

A particular distribution of spatio-temporal image derivatives arises at points which view arbitrary textures which always follow a constant optic flow. In this case, the image derivatives should fit the optic-flow constraint equation [16] $I_x u + I_y v + I_t = 0$, for an optic-flow vector (u, v) which remains constant through time.

Measurement

The 3-vector consisting of the x, y, t derivatives of the intensity is $\vec{m} = \langle I_x, I_y, I_t \rangle$.

Model parameters

The model parameters are the components of the optic-flow vector u, v.

Score

Any measurement arising from an object in the scene which satisfies the image brightness constancy equation and is moving with a velocity u, v will satisfy the optic-flow constraint equation: $I_x u + I_y v + I_t = 0$. The score for a given measurement $\vec{m}$ is the squared deviation from this constraint: $f(\vec{m}) = (I_x u + I_y v + I_t)^2$.

Estimation

For a given set of k background samples, the optic flow is determined by the solution to the following linear system (note that here the optic flow is assumed to be constant over time, not over space—the linear system uses the values of I_x, I_y, I_t for the same pixel in k different frames):

$$\begin{bmatrix} I_{x1} & I_{y_1} \\ I_{x2} & I_{y_2} \\ \vdots & \vdots \\ I_{xk} & I_{y_k} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{tk} \end{bmatrix}. \tag{6}$$

The solution to this linear system is the values of (u, v) which minimize the sum of the squared residual error. The mean squared residual error is a measure of how well this model fits the data, and can be calculated as follows:

$$\text{mean squared residual error} = \frac{\sum_{i=1,\dots,k} (I_{x_i} u + I_{y_i} + I_{ti})^2}{n}. \tag{7}$$

A map of this residual at every pixel is shown for a traffic intersection scene in Figure 2.

Online estimation

The above linear system can be solved using the pseudo-inverse. This solution has the following form:

$$\begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}. \tag{8}$$

The components of the matrices used to compute the pseudo-inverse can be maintained and updated with the measurements from each new frame. The best-fitting flow field for the "intersection" dataset is plotted in Figure 2.

4.7. Linear prediction based upon time history

The following model does not fit the spatio-temporal image processing paradigm exactly, but is included for the sake of comparison. The fundamental background model used in [2] was a one-step Wiener filter. This is linear predictor of the intensity at a pixel based upon the time history of intensity at that particular pixel. This can account for periodic variations of pixel intensity.

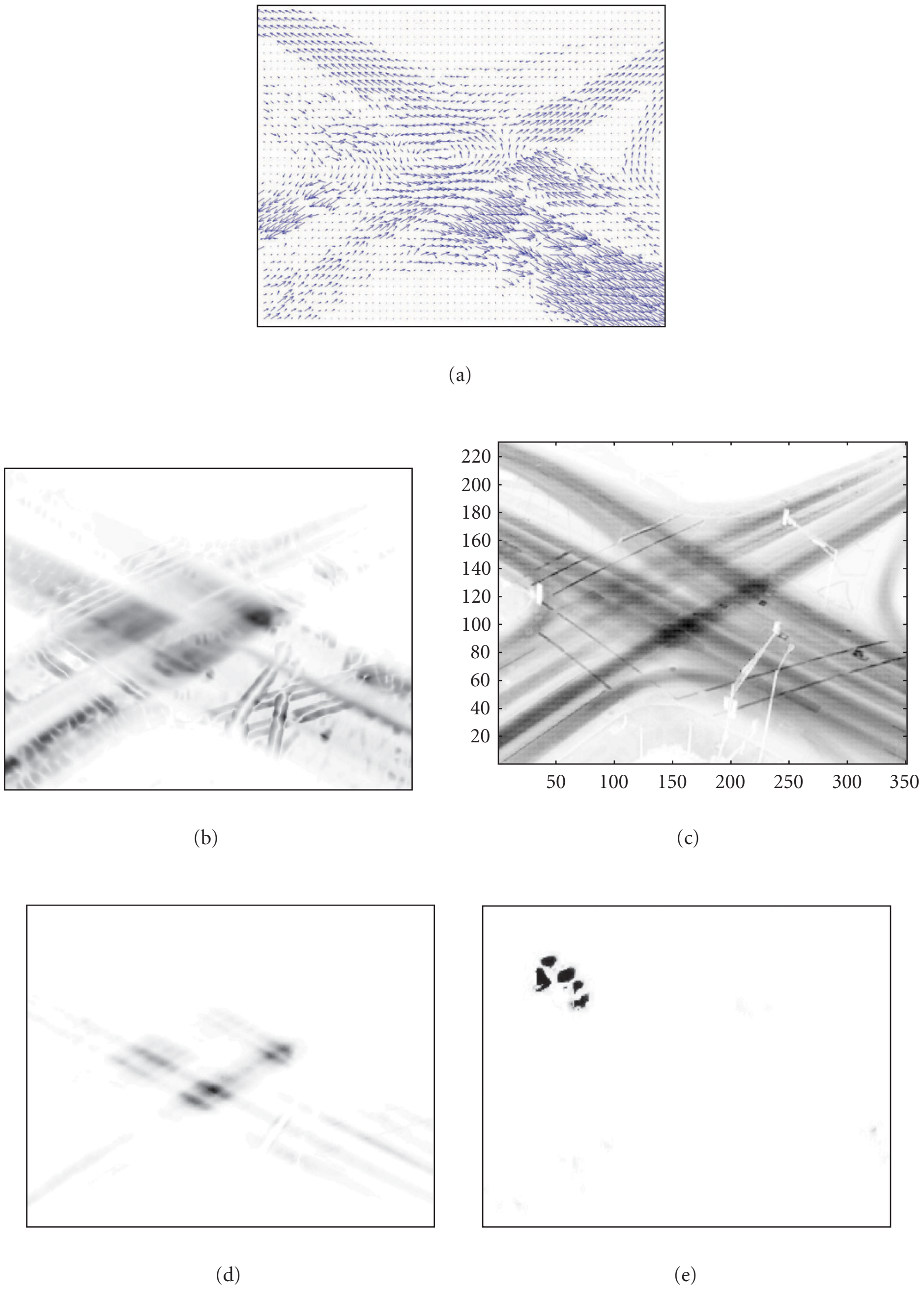

Figure 2: (a) The best-fitting optic-flow field, for a 19 000 frame video of a traffic intersection. (b) The residual error of fitting a single-optic-flow vector to all image derivative measurements at each pixel. (c) Residual error in fitting a single intensity value to each pixel. (d) Residual error in fitting a Gaussian distribution to the image derivative measurements. (e) The error function, when using the optic-flow model, of the intersection scene during the passing of an ambulance following a path not exhibited when creating the background model. The deviation scores are 3 times greater than the deviations for any car.

Measurement

The measurement includes two parts, the intensity at the current frame $I(t)$, and the recent time history of intensity values at a given pixel $I(t-1), I(t-2), \dots, I(t-p)$, so the complete measurement is $\vec{m} = \langle I(t), I(t-1), I(t-2), \dots, I(t-p)\rangle$.

Score

The estimation procedure gives a prediction $\hat{I}(t)$ which is calculated as follows:

$$\hat{I}(t) = \sum_{i=1\to p} a_i I(x, y, t-i). \tag{9}$$

Then the score is calculated as the failure of this prediction:

$$f(\vec{m}) = (I(t) - \hat{I}(t))^2. \tag{10}$$

Estimation

The best-fitting values of the coefficients of the linear estimator $(a_1, a_2, \dots, a_p)$ can be computed as the solution to the linear system defined as follows:

$$\begin{bmatrix} I(1) & I(2) & \cdots & I(p) \\ I(2) & I(3) & \cdots & I(p+1) \\ I(3) & I(4) & \cdots & I(p+2) \\ \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & I(n-1) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} I(p+1) \\ I(p+2) \\ I(p+3) \\ \vdots \\ I(n) \end{bmatrix} \tag{11}$$

Online estimation

The pseudo-inverse solution for the above least squares estimation problem has a $p \times p$ and a $1 \times p$ matrix with components of the form

$$\sum_i I(i)I(i+k), \tag{12}$$

for values of k ranging from 0 to $(p+1)$. These $p^2 + p$ components are required to compute the least squares solution. It is only necessary to maintain the pixel values for the prior p frames to accurately update all these components. More data must be maintained from frame to frame for this model than previous models. The amount of data is independent, however, of the length of the video input, so this fits with a model of streaming video processing.

5. EXPERIMENTAL RESULTS

We captured video imagery from a variety of natural scenes, and used the online parameter estimation processes to create a model of background motion. Each model produces a background score at each pixel for each frame. The mean squared deviation measure, calculated at each pixel, gives a picture of how well a particular model applies to different parts of a scene. Figure 2 shows the mean deviation function at each pixel for different background models.

By choosing a threshold, this background score can be used to classify that pixel as background or foreground. However, the best threshold depends upon the specific application. One threshold independent characterization of the performance of the classifier is a receiver operator characteristic (ROC) plot. The ROC plots give an indication of the tradeoffs between false positive and false negative classification errors for a particular pixel.

5.1. Receiver operator characteristic plots

ROC plots describe the performance (the "operating characteristic") of a classifier which assigns input data into dichotomous classes. An ROC plot is obtained by trying all possible threshold values, and for each value, plotting the sensitivity value (fraction of true positives correctly identified) on the y-axis against the (1-specificity) value (fraction of false positive identifications) on the x-axis. A classifier which randomly classifies input data will have an ROC plot which is a line of slope 1, and the optimal classifier (which never makes either a false positive or false negative error) is characterized by an ROC curve passing through the top left corner (0, 1), indicating perfect sensitivity and specificity (see Figure 3). The plots have been used extensively in evaluation of computer vision algorithm performance [17]. This study is a technology evaluation in the sense described in [18], in that it describes the performance characteristics for different algorithms in a comparative setting, rather than defining and testing an end-to-end system.

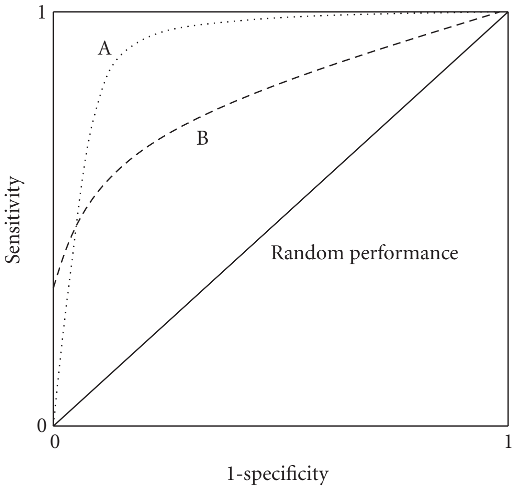

FIGURE 3: Receiver operator characteristic (ROC) curves describe the performance characteristics of a classifier for all possible thresholds [17, 19]. A random classifier has an ROC curve which is a straight line with slope 1. A curve like that labeled A has a threshold choice which defines a classifier which is both sensitive and specific. The nonzero y-intercept in the curve labeled B indicates a threshold exists where the classifier is somewhat sensitive, but gives zero false positive results.

These plots are defined for five models, each applied to four different scenes (shown in Figure 4) for the full length of the available data (300 frames for the tree sequences and 19 000 frames for the intersection sequence). Portions of the video clip with no unusual activity were selected by hand and background models were created from all measurements taken at that pixel, using the methods described in Section 4. Creating distributions for anomalous measurements was more difficult, because there was insufficient anomalous behavior at each pixel to be statistically meaningful and we lacked an analytic model of a plausible distribution of the anomalous measurements of image intensity and derivatives. Lacking an accepted model of the distribution of anomalous $\langle I, I_x, I_y, I_t \rangle$ measurements in natural scenes, we choose to generate anomalous measurements at one pixel by sampling randomly from background measurements at all other locations (in space and time) in every video tested.

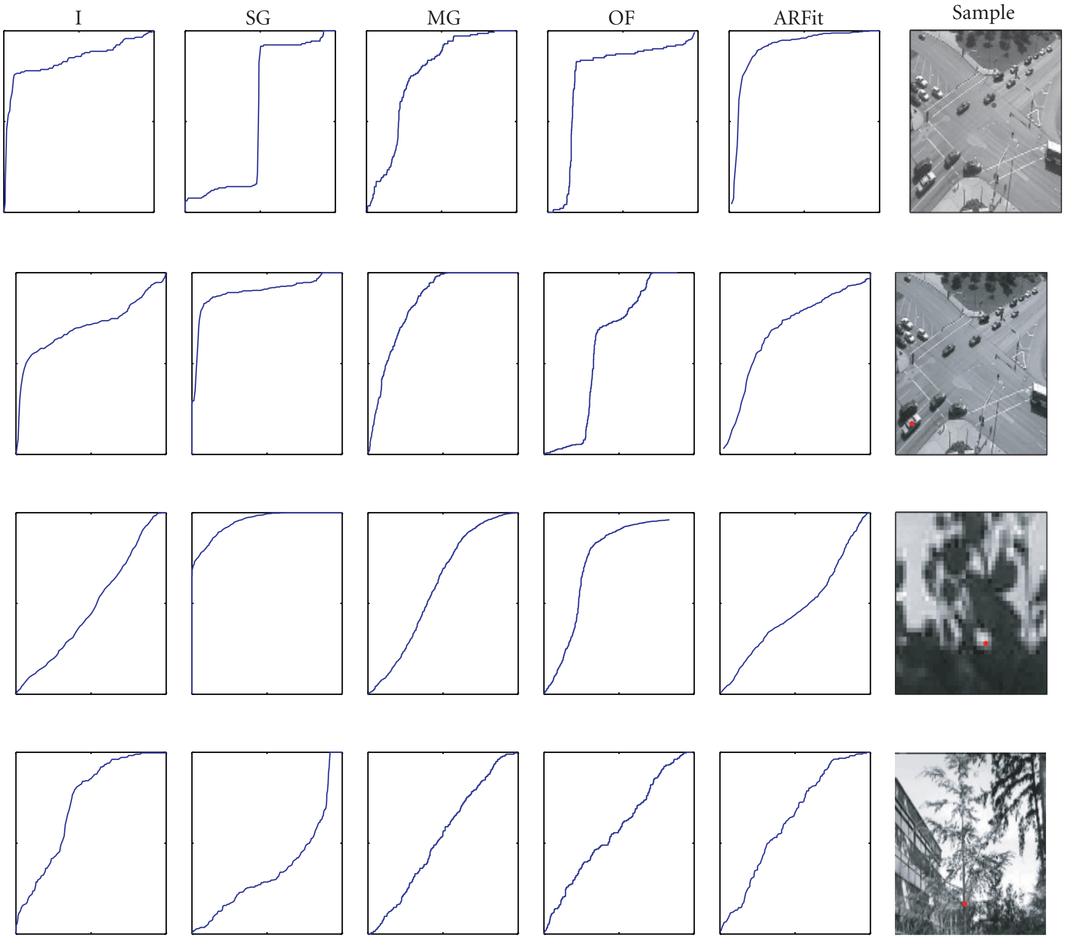

Figure 4: Each ROC plot represents the trade-offs between the sensitivity of the classifier on the (y-axis), and 1-specificity on the x-axis. The model is defined at one pixel (x, y position marked by dots on each image), and plots are shown for a model based upon (I) intensity, (SG) Gaussian distribution in (I, I_x, I_y, I_t)-space, (MG) multiple Gaussian, (OF) optic flow, and (ARfit) linear prediction based upon intensity in prior frames. The comparison between the first and second rows shows that all models perform better on parts of the intersection with a single direction of motion rather than a point that views multiple motions, except the auto-regressive model (from [2]), for which we have no compelling explanation for its excellent performance. The third and fourth rows compare the algorithms viewing a tree branch, the top is a branch moving slowly in the wind, the bottom (a dataset from [2]), is being shaken vigorously. For the third row, the multiple-Gaussian model is the basis for a highly effective classifier, while the high speed and small features of the data set on the fourth row make the estimation of image derivatives ineffective, so all the models perform poorly.

The ROC plots are created by using a range of different threshold values. For each model, the threshold value defines a classifier, and the sensitivity and specificity of this classifier are determined using measurements drawn from our distribution. The plot shows, for each threshold, 1-specificity versus sensitivity. Each scene illustrated in Figure 4 merits a brief explanation of why the ROC plot for each model takes the given form.

(i) The first scene is a traffic intersection, and we consider the model for a pixel in the intersection that sees two directions of motion. The intensity model and the single Gaussian effectively compare new data to the color of the pavement. The multiple-Gaussian model has very poor performance (below chance for some thresholds). There is no single-optic-flow vector which characterizes the background motions.

(ii) The second scene is the same intersection, but we consider a pixel location which views objects with a consistent motion direction. Both the multiple-Gaussian and the multiple-optic-flow models have sufficient expressive power to capture the constraint that the motion at this point is consistently in one direction with different speeds.

(iii) The third scene is a tree with leaves waving naturally in the wind. The model which uses EM to fit a collection of Gaussians to this data is clearly the best, because it is able to specify correlations between the image gradient and the image intensity (it can capture the specific

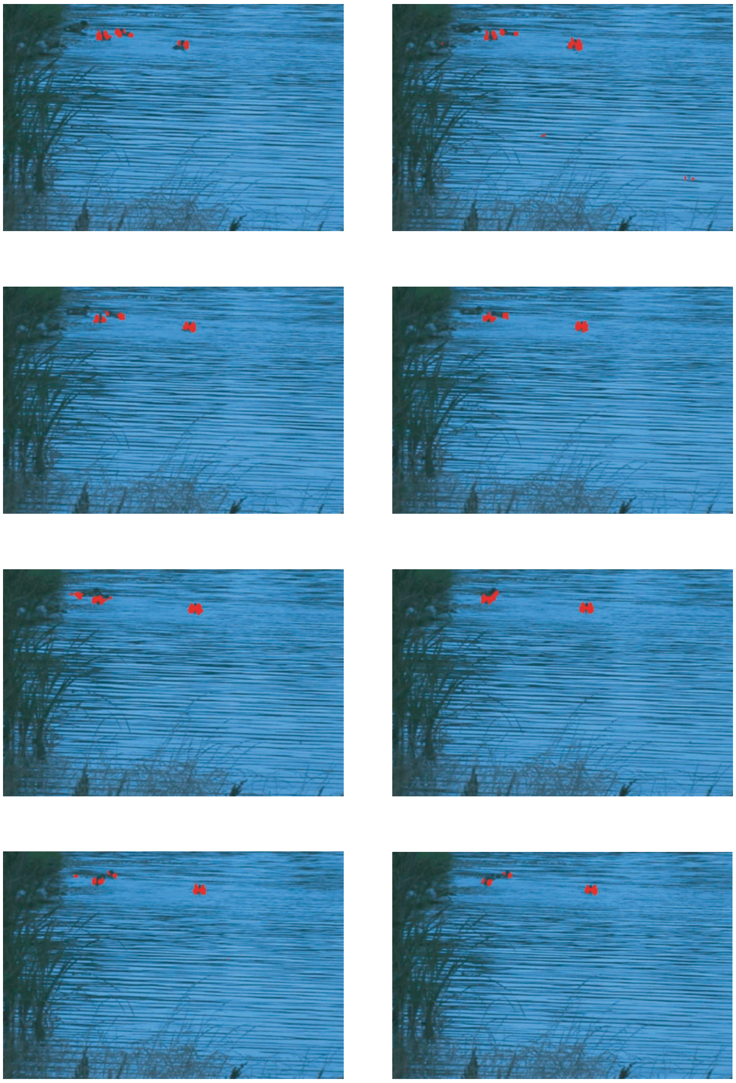

Figure 5: Every tenth frame of a video of ducks swimming over a lake with waves and reeds moving in the wind. Marked in red are pixels for which the likelihood that spatio-temporal filter responses arose from the background model fell below a threshold. These responses are from a single set of spatio-temporal filter measurements, that is, no temporal continuity was used to suppress noise. The complete video is available at http://www.cse.wustl.edu/~pless/ind.html.

changes of a leaf edge moving left, a leaf edge moving right, the static leaf color, and the sky). The motions do not corresponds to a small set of optic-flow vectors, and are not effectively predicted by recent time history.

(iv) The final test is the tree scene from [2], a tree which was vigorously shaken from just outside the field of view. The frame-to-frame motion of the tree is large enough that it is not possible to estimate accurate derivatives, making spatio-temporal processing inappropriate.

5.2. Real-time implementation

Except for the linear prediction based upon time history, each of the above models has been implemented on a fully real-time system. This system runs on an 800 MHz Sony Vaio laptop with a Sony-VL500 firewire camera. The system is based on Microsoft Direct X and therefore has a great deal of flexibility in camera types and input data sources. With the exception described below, the system runs at 640-by-480 resolution at 30 fps, for all models described in the last section. The computational load is dominated by the image smoothing and the calculation of image derivatives.

Figure 5 shows the results of running this real-time system on a video of a lake with moving water and reeds moving in the wind. Every tenth frame of the video is shown, and independent pixels are marked in red. The model uses a single Gaussian to represent the distribution of the measurement vectors at each pixel, and updates the models to overweight the newest data, effectively making the background model dependent primarily on the previous 5 seconds. The fifth, sixth, and seventh frames shown here indicate the effect of this. The duck in the top left corner remained stationary for the first half of the sequence. When the duck moves, the water motion pattern is not initially represented in the background model, but by the eighth frame, the continuous updates of the background model distribution have incorporated the appearance of the water motion.

The multiple-Gaussian model most often performed best in the quantitative studies. However, iterative expectation maximization algorithm requires maintaining all the training data, and is therefore not feasible in a streaming video context. Implementing the adaptive mixture models exactly as in [20] (although their approach was modeling a distribution of a different type of measurements) is a feasible approach to creating a real-time system with similar performance.

The complete set of parameters required to implement any of the models defined in Section 4 are the choice of the model, image blurring filter, exponential forgetting factor (over-weighting the newest data, as discussed above), and a threshold to interpret the score as a classifier. The optimal image blurring factor and the exponential forgetting factor depend on the speed of typical motion in the scene, and the period over which motion patterns tend to repeat—for example, in a video of a traffic intersection, if the forgetting factor is too large, then every time the light changes, the motion will appear anomalous. The choice of model can be driven by the same protocol used in the experimental studies, as the only human input is the designation of periods of only background motion. However, to be most effective, the choice of foreground distribution should reflect any additional prior knowledge about the distribution of image derivatives for anomalous objects that may be in the scene.

6. CONCLUSION

The main contributions of this paper are the presentation of the image derivative models of Sections 4.4 and 4.5, which are, to the authors knowledge, the first use of the distribution of spatio-temporal derivative measurements as a background model, as well as the optic-flow model of Section 4.6, which introduces new techniques for online estimate of the optic flow at a pixel that best fits image derivative data collected over long time periods. Additionally, we have presented a framework which allows the empirical comparison of different models of dynamic backgrounds.

This work focuses on the goal of expanding the set of background motions that can be subtracted from video imagery. Automatically ignoring common motions in natural outdoor and pedestrian or vehicular traffic scenes would improve many surveillance and tracking applications. It is possible to model much of these complicated motion patterns with a representation which is local in both space and time and efficient to compute, and the ROC plot gives evidence for which type of model may be best for particular applications. The success of the multiple-Gaussian model argues for further research in incremental EM algorithms which fit in a streaming video processing model.

REFERENCES

[1] H. Farid and E. P. Simoncelli, "Optimally rotation-equivariant directional derivative kernels," in *Proc. 7th International Conference on Computer Analysis of Images and Patterns (CAIP '97)*, pp. 207–214, Kiel, Germany, September 1997.

[2] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proc. 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 1, pp. 255–261, Kerkyra, Greece, September 1999.

[3] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proc. IEEE International Conference on Computer Vision (ICCV '99) FRAME-RATE Workshop*, Kerkyra, Greece, September 1999.

[4] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, vol. 2, pp. 246–252, Fort Collins, Colo, USA, June 1999.

[5] I. Haritaoglu, D. Harwood, and L. Davis, "W4S: A real time system for detecting and tracking people in 2.5 D," in *Proc. 5th European Conference on Computer Vision (ECCV '98)*, pp. 887–892, Freiburg, Germany, June 1998.

[6] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 774–780, 2000.

[7] R. Pless, T. Brodsky, and Y. Aloimonos, "Detecting independent motion: The statistics of temporal continuity," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 768–773, 2000.

[8] F. Liu and R. W. Picard, "Finding periodicity in space and time," in *Proc. 6th International Conference on Computer Vision (ICCV '98)*, pp. 376–383, Bombay, India, January 1998.

[9] S. A. Niyogi and E. H. Adelson, "Analyzing and recognizing walking figures in XYT," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94)*, pp. 469–474, Seattle, Wash, USA, June 1994.

[10] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis and applications," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 781–796, 2000.

[11] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust kalman filter," in *Proc. 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 1, pp. 44–50, Nice, France, October 2003.

[12] S. Soatto, G. Doretto, and Y. N. Wu, "Dynamic textures," in *Proc. International Conference on Computer Vision (ICCV '98)*, pp. 439–446, Bombay, India, January 1998.

[13] R. P. Wildes and J. R. Bergen, "Qualitative spatiotemporal analysis using an oriented energy representation," in *Proc. 6th European Conference on Computer Vision (ECCV '00)*, pp. 768–784, Dublin, Ireland, June–July 2000.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[15] M. Aitkin and D. B. Rubin, "Estimation and hypothesis testing in finite mixture models," *Journal of the Royal Statistical Society B*, vol. 47, no. 1, pp. 67–75, 1985.

[16] B. K. P. Horn, *Robot Vision*, McGraw-Hill, New York, NY, USA, 1986.

[17] K. W. Bowyer and P. J. Phillips, Eds., *Empirical Evaluation Techniques in Computer Vision*, IEEE Computer Society Press, Santa Barbara, Calif, USA, 1998.

[18] P. Courtney and N. A. Thacker, "Performance characterisation in computer vision: the role of statistics in testing and design," in *Imaging and Vision Systems: Theory, Assessment and Applications*, J. Blanc-Talon and D. Popescu, Eds., NOVA Science Books, Huntington, NY, USA, 1993.

[19] J. P. Egan, *Signal Detection Theory and ROC Analysis*, Academic Press, New York, NY, USA, 1975.

[20] M. Harville, G. G. Gordon, and J. Woodfill, "Foreground segmentation using adaptive mixture models in color and depth," in *Proc. IEEE Workshop on Detection and Recognition of Events in Video*, pp. 3–11, Vancouver, British Columbia, Canada, July 2001.

Robert Pless is an Assistant Professor of computer science at Washington University, where he cofounded the Media and Machines Laboratory. Dr. Pless holds a B.S. degree from Cornell University and a Ph.D. degree from the University of Maryland, both in computer science. Dr. Pless has a research focus on video analysis, especially data-driven algorithms for video surveillance and nonrigid motion understanding. He served as Chairman of the 2003 IEEE International Workshop on Omni-directional Vision and Camera Networks. Dr. Pless also serves as Assistant Director of the Center for Security Technologies, an interdisciplinary center including 45 faculty members from 4 different schools of Washington University, which concentrates on both fundamental research in sensors and algorithms and the interplay between security technologies, privacy, policy, and ethics.