Evaluation of Local Models of Dynamic Backgrounds

Robert Pless, John Larson, Scott Siebers, and Ben Westover
Department of Computer Science and Engineering
Washington University in St. Louis
{pless,jplarson,sgs2,ben}@cse.wustl.edu

Abstract

Background subtraction is the first step of many video surveillance applications. What is considered background varies by application, and may include regular, systematic, or complex motions. This paper explores the use of several different local *spatio-temporal models of a background, defined at each pixel in the image. We present experiments with real image data and conclude that appropriate local representations are sufficient to make background models of complicated real world motions. Empirical studies illustrate, for example, that an optical flow-based model is able to detect emergency vehicles whose motion is different from those typically observed in traffic scenes. We conclude that "different models are appropriate for different scenes", but give criteria by which one can choose which model will be best.*

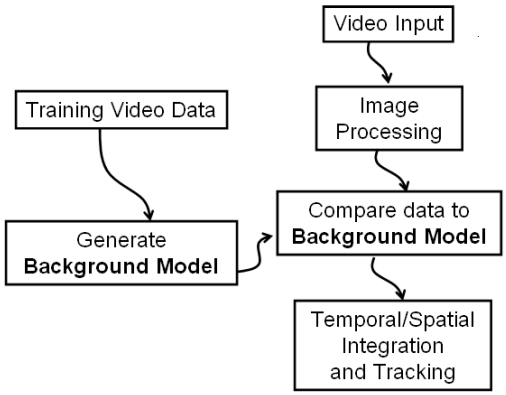

Figure 1. The generic framework of the front end of visual surveillance systems. This work focusses on exploring different local background models.

1 Introduction

Video surveillance applications seek to find, identify, or track objects and events that appear before the camera. A first step in many surveillance algorithms is background subtraction — identifying the background regions of the image or video which should be ignored. This allows resources to be devoted to tracking or interpreting the remaining data. What should be considered "background" is an application specific question. In this paper we aim to extend the set of motions within video scenes that can be considered background. We propose and analyze background representations which effectively model natural scenes including waving grass or trees and intersections with cars moving along varied but regular trajectories. The ability to model complicated background motions will allow surveillance and tracking algorithms to be deployed in a wider variety of scenes.

The background models are computed with techniques characterized as spatio-temporal image processing. 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 its spatial and temporal derivatives, measured at each pixel in each frame. Unlike interest points or features, these measurements are defined at every pixel in the video sequence. With appropriate blurring, these derivatives give robust measurements to form a basis for further processing. Optimality criteria and algorithms for creating derivative and blurring filters of a particular size have been developed by [2], and lead to significantly better results than estimating derivatives by applying Sobel filters to raw images.

The framework of many surveillance systems is shown in Figure 1. These 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 [4] (e.g. the use of blue screens in the entertainment industry), or (b) consistent motions, where the image is static [3] or undergoing a global transformation which can be affine [8] or planar projective [5].

Each background model defines an error measure. The analysis of new video data consists of calculating this error for each pixel in each frame. This measure of error is 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 [6].

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 is defined independently for each pixel (x, y) in the scene, and is based upon the distribution of the image intensity $I(x, y)$ and its spatial $I_x(x, y), I_y(x, y)$, and temporal derivatives $I_t(x, y)$ at that pixel. Qualitative analysis of local image changes have been carried out using oriented energy measurements [7], here we look at the quantitative predictions that are possible with similar representations of image variation.

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. Every pixel in every frame has an image measurement vector of the form $\langle I, I_x, I_y, I_t \rangle$; a complete background model includes a function which returns a score which is smaller when the pixel fits the background model.

2 Models of background motion

Each local model of image variation is defined with four parts. First, the measurement – which part of the spatio-temporal image that 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 updating for each new frame of data within the context of streaming video applications.

2.1 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 of the just 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 m arises from the background model is $f(\vec{m}) = \frac{(\vec{m} - \vec{m}_{\text{background}})^2}{\sigma^2}$. 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 $\vec{m}_{background}$ is assumed to be known a-priori.

2.2 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, $\vec{m} = \langle r, g, b \rangle$.

Independence 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 which maintains a count n of the number of background frames and the current estimate of m_μ. This estimate can be updated: $\vec{m}_{\mu_{new}} = \frac{n-1}{n}\vec{m}_\mu + \frac{1}{n}\vec{m}$.

2.3 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 m arises from the background model is $f(\vec{m}) = \frac{||\vec{m} - \vec{m}_\mu||_2^2}{\sigma^2}$.

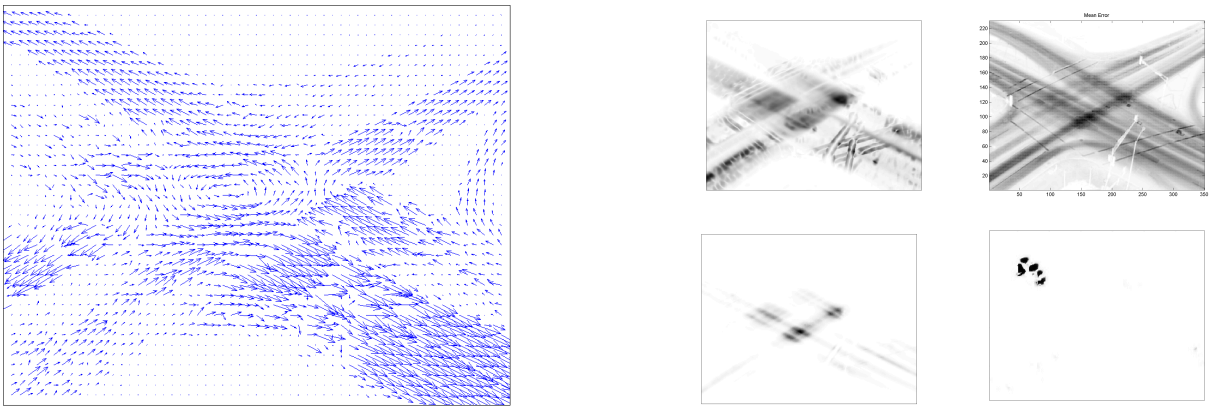

Figure 2. (Left) The best fitting optic flow field, for a 19,000 frame video of a traffic intersection. (Top Middle) The residual error of fitting a single optic flow vector to all image derivative measurements at each pixel. (Top Right) Residual error in fitting a single intensity value to each pixel. (Bottom Middle) Residual error in fitting a Gaussian distribution to the image derivative measurements. (Bottom Right) The error function, when using the optic flow model, of the intersection scene during the passing of an ambulance that was not when creating the background model. The deviation scores are 3 times greater than the deviations for any car.

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 straight-forward 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: $\vec{m}_{\mu_{new}} = \frac{1}{n+1}\vec{m} + \frac{n}{n+1}\vec{m}_\mu$. If each measurement is assumed to have variance 1, the variance σ^2 is updated as follows: $\sigma^2_{new} = (\frac{1}{\sigma^2} + 1)^{-1}$.

2.4 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: $\vec{m} = \langle I, I_x, I_y, I_t \rangle$.

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: $f(\vec{m}) = (\vec{m} - \vec{m}_\mu)^\top \Sigma^{-1} (\vec{m} - \vec{m}_\mu)$

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

$$\vec{m}_\mu = \frac{\sum_{i=1\ldots k} m_i}{k}$$

$$\Sigma = \frac{\sum_{i=1\ldots k} (m_i - \vec{m}_\mu)(m_i - \vec{m}_\mu)^\top}{k-1}.$$

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:

$$\Sigma_{new} = \frac{n}{n+1}\Sigma + \frac{n}{(n+1)^2}(\vec{m} - \vec{m}_\mu)(\vec{m} - \vec{m}_\mu)^\top.$$

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

Using several multi-dimensional 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 3 use three) multi-dimensional Gaussian distributions to the measurements at a particular pixel location.

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:

$$f(\vec{m}) = min_i(\vec{m} - \vec{m}_{\mu_i})^\top \Sigma_i^{-1}(\vec{m} - \vec{m}_{\mu_i})$$

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.

2.6 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: $I_x u + I_y v + I_t = 0$, 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: $\vec{m} = \langle I_x, I_y, I_t \rangle$.

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 best fitting solution for the optic flow is determined by the solution to the linear system (note that here the optic flow is assumed to be constant over time, not over space — each of the background measurements consist of the values of I_x, I_y, I_t for the same pixel in k different frames):

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xk} & I_{yk} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{tk} \end{bmatrix}$$

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

$$\sigma^2 = \frac{\sum_{i=1\ldots k} (I_{x_i} u + I_{y_i} + I_{t_i})^2}{n}.$$

A map of this residual at every pixel is shown in Figure 2.

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

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

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

2.7 Linear Prediction based upon time history

The following model 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 [6] 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.

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 $I(t-1), I(t-2), \ldots, I(t-p)$, so the complete measurement is $\vec{m} = \langle I(t), I(t-1), I(t-2), \ldots, I(t-p) \rangle$.

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

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

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

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

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

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

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:

$$\sum_i I(i) I(i+k),$$

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 then 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.

3 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 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 classifications errors for a particular pixel.

3.1 Receiver Operator Characteristic (ROC) 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. This study is a technology evaluation in the sense described in [1], in that it describes the performance characteristics for different algorithms in a comparative setting, rather than defining and testing an end to end system.

These plots are defined for five models, each applied to four different scenes (shown in Figure 4). The y-axis of each plot is the sensitivity: the probability that a measurement from the background is correctly classified as background. The x-axis of each plot is (1-specificity): the probability that a measurement *not* from the background is classified as background. Lacking an accepted model of the distribution of $\langle I, I_x, I_y, I_t \rangle$ measurements in natural scenes, we choose to sample randomly from every location (in space and time) in every video tested.

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:

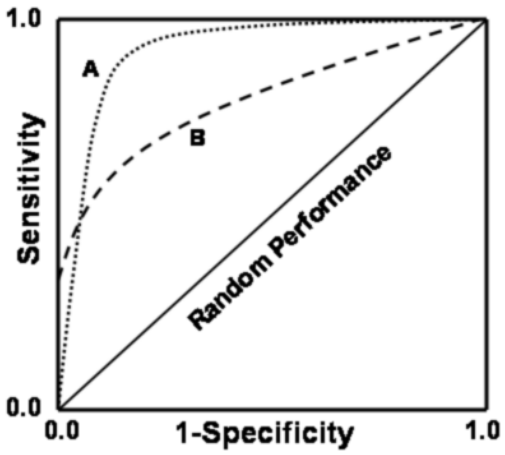

Figure 3. Receiver Operator Characteristic (ROC) curves describe the performance characteristics of a classifier for all possible thresholds. A random classifier has a ROC curve which is a straight line with slope 1. A curve like that labelled (A), has a threshold choice which defines a classifier which is both sensitive and specific. The non-zero y-intercept in the curve labelled (B) indicates a threshold exists where the classifier is somewhat sensitive, but gives zero false positive results.

- 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.

- 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.

- 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

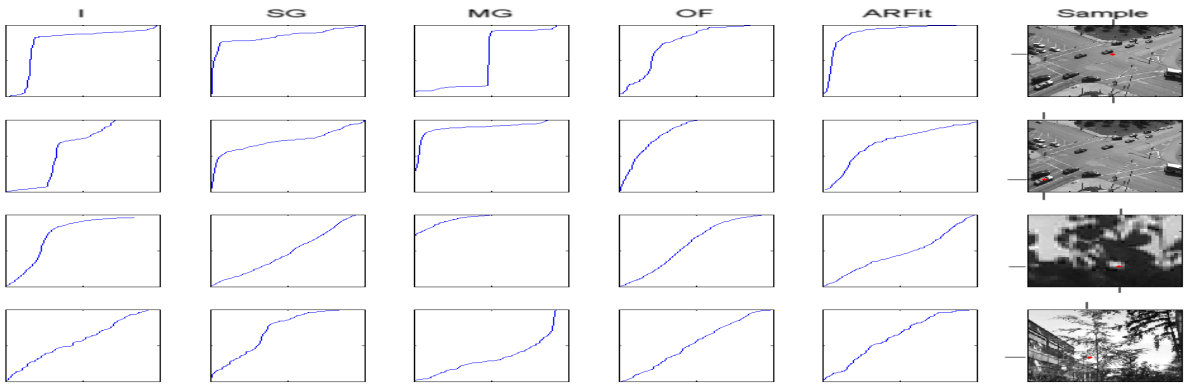

Figure 4. Each ROC plot represents the tradeoffs between the sensitivity of the classifier on the (y-axis), and (1-specificity) on the (x-axis). The model is defined at one pixel (x,y position marked by the hashes on the axes), and plots are shown for a model based upon: (I) intensity, (SG) Gaussian 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.

is able to specify correlations between the image gradient, and the image intensity (it can capture the specific 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.

- The final test is the tree scene from [6], 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.

Finally, included with this submission is a video showing a brief pair of clips. First a clip including an ambulance, followed by the deviation function for this clip, and second a clip of regular traffic flow, followed by the deviation function for that part of the video. One frame of the deviation including the ambulance is shown in Figure 2.

4 Conclusion

This work focusses 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-Gaussians model argues for research in incremental EM algorithms which fit in a streaming video processing model.

References

[1] P. Courtnet and N. A. Thacker. Performance characterisation in computer vision: The role of statistics in testing and design. In J. Blanc-Talon and D. Popescu, editors, *Imaging and Vision Systems: Theory, Assessment and Applications*. NOVA Science Books, 1993.

[2] H. Farid and E. P. Simoncelli. Optimally rotation-equivariant directional derivative kernels. In *Computer Analysis of Images and Patterns (CAIP)*, 1997.

[3] I. Haritaoglu, D. Harwood, and L. Davis. W4s: A real time system for detecting and tracking people in 2.5 d. In *ECCV*, 1998.

[4] 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*, 1999.

[5] 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.

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

[7] R. P. Wildes and J. R. Bergen. Qualitative spatiotemporal analysis using an oriented energy representation. In *Proc. European Conference on Computer Vision*, pages 768–784, 2000.

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