SANDIA REPORT
SAND2009-5546
Unlimited Release
Printed August 2009

Robust Real-Time Change Detection in High Jitter

Katherine M. Simonson and Tian J. Ma

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2009-5546
Unlimited Release
Printed August 2009

Robust Real-Time Change Detection in High Jitter

Katherine M. Simonson and Tian J. Ma
Data Analysis & Data Exploitation Department
Sandia National Laboratories
Albuquerque, NM 87185-1208

Abstract

A new method is introduced for real-time detection of transient change in scenes observed by staring sensors that are subject to platform jitter, pixel defects, variable focus, and other real-world challenges. The approach uses flexible statistical models for the scene background and its variability, which are continually updated to track gradual drift in the sensor's performance and the scene under observation. Two separate models represent temporal and spatial variations in pixel intensity. For the temporal model, each new frame is projected into a low-dimensional subspace designed to capture the behavior of the frame data over a recent observation window. Per-pixel temporal standard deviation estimates are based on projection residuals. The second approach employs a simple representation of jitter to generate pixelwise moment estimates from a single frame. These estimates rely on spatial characteristics of the scene, and are used gauge each pixel's susceptibility to jitter. The temporal model handles pixels that are naturally variable due to sensor noise or moving scene elements, along with jitter displacements comparable to those observed in the recent past. The spatial model captures jitter-induced changes that may not have been seen previously. Change is declared in pixels whose current values are inconsistent with both models.

This work performed under sponsorship of the
Laboratory Directed Research and Development (LDRD) Program

Contents

List of Tables

List of Figures

1 – INTRODUCTION

Staring remote sensors operate across a wide range of viewing geometries, wavelengths, frame rates, pixel resolutions, and areas of application. Fixed, ground-based cameras can be used for perimeter security and intrusion detection, traffic monitoring, military route surveillance, and border protection. Airborne or space-based sensors might be employed for environmental land use monitoring, weather prediction and storm tracking, and ballistic missile defense. The rapid detection of new activity in a sensor's field of view can be a critical processing step in each of these scenarios. *Change detection* is thus an active area of research within the image science and engineering community, and many authors have contributed methods and approaches to the basic problem of identifying "interesting" changes to an imaged scene. The interest generated by various types of change is generally application-specific: motion of a wind sock may be considered "signal" when determining flight conditions along a runway, and "noise" for perimeter security. For this reason, the various algorithms and approaches are usually designed for (or tuned to) specific problem types.

The article by Radke et al. [20] provides a nice overview and taxonomy of the extensive literature in the field of image change detection. The expression *background modeling* is often used for problems like the one to be considered here, in which a sequence of frame-rate video data is available, and the changes to be detected occur over a matter of seconds. Many authors working on this type of problem assume that the frame sequence is perfectly registered [23, 26 30], so that intensity changes due to camera jitter are absent. For non-stationary cameras, registration of the current frame to a previous frame, estimated background, or mosaic image is often recommended [16, 24]. However, registration to a small fraction of a pixel may be required to mitigate jitter effects [25], and this degree of precision is not always feasible, particularly at higher sensor frame rates [22].

Comparatively few authors have addressed the problem of change detection when precise frame registration is neither assumed nor achievable. Bruzzone and Cossu [3] develop a non-parametric estimate of the "registration noise" distribution for a single frame, and determine that change has occurred when pixels show intensity differences significantly larger than those

predicted by the estimated distribution. This computationally intensive approach is best suited to offline analysis of selected frame pairs, rather than real-time change detection. Elgammal et al. [8] apply kernel density estimation to a temporal frame history and compute pixel density estimates that can incorporate both illumination and motion-induced changes in intensity. Change is detected when a pixel's current observed intensity is improbable according to the historical model. By contrast, in [12] two separate probability density functions (PDFs) are trained per pixel, from a single prior frame. The two PDFs are designed to account for both uncorrelated mean-zero noise (a single Gaussian distribution), and jitter-induced noise that is correlated both temporally and spatially (a mixture of Gaussians). Detection occurs when a pixel's observed intensity is statistically inconsistent with *both* models. Finally, Jodoin et al. [11] perform change detection in high-jitter sequences by differencing average and observed pixel activity maps that can be updated adaptively.

The method introduced in this paper is designed for staring sensors with a frame rate that precludes precise registration, and for which real-time, causal change detection is required. Sensor or platform jitter is expected, which may exhibit a non-stationary distribution featuring sudden increases in jitter variance, as when a nearby motor is energized and induces additional vibration of the sensor. The sensor's pointing may slowly degrade, so that its field of view gradually drifts away from the initial scene. The scene itself may be subject to natural change, due to wind-driven vegetation or water surface motion at selected pixel locations. The target changes to be detected may be quite dim in comparison to the scene gradients and the natural scene variability. And finally, no *a priori* knowledge of the scene content (e.g. site model) is available, other than a brief time history of frames used to initialize the detector.

The approach was originally designed for cases in which true change is *rare* (the majority of frames do not contain "interesting" change) and *small* (the changes of interest impact a tiny percentage of pixels, and are often not readily detectable by a human observer). Nonetheless, it has proven quite effective in other scenarios, with appropriate modifications to the initialization procedure. Our current implementation is limited to single-band (greyscale) data.

Similar to [12], our approach to designing a detector employs two separate models. However, we estimate only the first two moments of each underlying distribution, and introduce

new techniques for generating and updating the moment estimates at sensor frame rates. The two models are designed to capture both the temporal and spatial characteristics of each pixel's behavior. Together, they provide a change detection capability that is robust to pixels that are inherently variable due to wind-induced motion of vegetation, water surface changes, or sensor electronics; and pixels that are subject to correlated, jitter-induced changes in intensity, which may or may not be observed during the initialization period.

The overall framework for detection is outlined in the next section. In Section 3, the temporal approach to characterizing scene background is discussed, while a novel spatial method for estimating pixel intensity means and variances is introduced in Section 4. The parameters used to tune the general change detection approach to specific applications are described in Section 5. Examples illustrating detection in several challenging scenarios are provided in Section 6, and the paper concludes with a discussion of ongoing and future work.

2 – DETECTION FRAMEWORK

At each discrete frame time, t, we capture a new image of dimension $NROW$ rows and $NCOL$ columns, where the total number of pixels per frame is $N = NROW \times NCOL$. For each pixel (k,h) in the frame corresponding to time t, our goal is to determine whether the intensity of the pixel at time t is consistent with current spatial and temporal models. The determination is made based on simple normalized differences:

$$Z_{TEMPORAL}(k,h;t) = \frac{x(k,h;t) - b_{TEMPORAL}(k,h;t)}{\xi_{TEMPORAL}(k,h;t-1)} \quad , \tag{1a}$$

$$Z_{SPATIAL}(k,h;t) = \frac{x(k,h;t) - b_{SPATIAL:L}(k,h;t)}{\xi_{SPATIAL}(k,h;t-1)} \quad . \tag{1b}$$

Here, $x(k,h;t)$ is the value observed in pixel (k,h) at time t, with $b_{TEMPORAL}$ and $b_{SPATIAL}$ representing the estimated background intensity levels for the temporal and spatial models, respectively. The corresponding standard deviation estimates, based on data observed prior to

time *t*, are $\xi_{TEMPORAL}$ and $\xi_{SPATIAL}$. Estimation of these quantities is covered in Sections 3 and 4.

Consistency with either model is judged by comparing the normalized absolute difference, $|z|$, to a fixed threshold denoted T_1, which does not vary with *k*, *h*, or *t*. If one is willing to assert that the distribution of normalized differences is Gaussian, then the threshold T_1 may be associated with a specific false alarm rate, and the decision may be formulated as a two-sided statistical hypothesis test. In some instances, it may be desirable to report only changes in a single direction. For example, with a thermal infrared sensor, only increases in heat may be of interest. When this is the case, *z* itself (or –*z*), rather than its absolute value, is compared to threshold T_1, and the significance level of the test is calculated using a one-sided measure.

Detection occurs at pixels whose observed intensities are judged to be inconsistent with *both* the temporal and spatial models. That is, whenever:

$$z_{MIN}(k,h;t) = \min\left\{\left|z_{TEMPORAL}(k,h;t)\right|,\ \left|z_{SPATIAL}(k,h;t)\right|\right\} > T_1 \quad . \tag{2}$$

If the targets of interest are expected to cover more than a single pixel, the false alarm rate may be reduced by requiring detection across a minimum number, *NEIGH*, of connected pixels prior to declaring that change has occurred. This condition can be tested quite efficiently [31].

3 – TEMPORAL MODELING

3.1 – Adaptive Subspace Projection

As its name implies, the temporal modeling approach relies on a running time history of each pixel's observed intensity. Subspace projection is used to capture the structure inherent in the highly correlated frame history data in a reasonably compact manner.

Several authors [2, 10, 14] have recommended the use of subspace projection methods to estimate (and then mitigate) the effects of line-of-sight (LOS) jitter on change detection. Barry and Klop [2] show that, apart from measurement noise, pixel output vectors lie within a two-

dimensional subspace determined by the LOS displacement vectors. This subspace can be estimated from a sequence of frames using the first two eigenvectors of the mean-centered data covariance matrix. Kirk and Donofrio [14] demonstrate jitter suppression by projecting observed frames into an estimated background subspace spanned by 16 basis vectors. And Diani et al. [7] use subspace projection to model a background that is non-stationary in both space and time. Most of these authors comment on the computational complexity of principal subspace estimation, which has limited its application for real-time background suppression.

Over the last few years, tremendous advances have been made in the development of fast and numerically stable techniques for adaptive principal subspace estimation. Many papers have appeared in journals dedicated to applied mathematics [18], signal processing [19], and neural networks [21]. Fortunately, a few authors [5, 6] include thorough literature reviews, which summarize the key approaches and contributions. Adaptive subspace estimation has been successfully applied to the detection of change in highly dynamic natural scenes [17]. For application to jitter suppression, we considered several different techniques. In [6], algorithms are classified by desired output (signal subspace or noise subspace), computational complexity, the number of parameters to be tuned, and whether or not the computed basis vector estimates are orthogonal at each iteration. Based on these criteria, we selected the Fast Approximate Power Iteration (FAPI) algorithm [1]. This approach provides orthonormal basis vector estimates at every iteration and is very efficient computationally, with complexity $O(NR)$, where N is the dimensionality of the data and R the size of the desired subspace.

For jitter suppression, we wish to estimate the dominant subspace spanned by the covariance matrix, $\boldsymbol{C}_{XX}(t)$, of a sequence of N-dimensional data vectors, $\{X(t), t \in \mathbb{Z}\}$. Here, $X(t)$ is the $N\times 1$ column vector obtained by vectorizing the $NROW\times NCOL$ image frame observed at time t. If the data are windowed exponentially, the covariance matrix at time t_0 would be estimated using:

$$\boldsymbol{C}_{XX}(t_0) = \sum_{u=-\infty}^{t_0} \beta_1^{t-u} X(u) X^T(u) \quad . \tag{3}$$

From an initial estimate, $\boldsymbol{C}_{XX}$ could be recursively updated at time $t > t_0$ as follows:

$$\boldsymbol{C}_{XX}(t) = \beta_1 \boldsymbol{C}_{XX}(t-1) + X(t)X^T(t) \quad . \tag{4}$$

Note that the data vectors are not mean-centered, so that the first eigenvector of $\boldsymbol{C}_{XX}$ will represent the mean frame. The tunable parameter β_1 lies in the range [0, 1] and governs the relative weight applied to the current basis vectors with each update. In our implementation, we have added one additional parameter, β_2, which slows the rate at which large apparent changes are incorporated into the background estimate, and will be discussed shortly. FAPI provides an efficient means to estimating the dominant subspace of the time-varying covariance matrix, without the computational cost of explicitly computing and decomposing $\boldsymbol{C}_{XX}$. At every iteration, the R current basis vectors are stored in an $N \times R$ weighting matrix, denoted $\boldsymbol{W}(t)$, which contains the estimates that are current once the observation at time t has been incorporated. The steps by which $\boldsymbol{W}(t\text{-}1)$ is updated to produce $\boldsymbol{W}(t)$, with decay parameter β_1, are clearly set out in [1].

Badeau et al. [1] initialize the FAPI basis vector estimates with a weight matrix whose first R rows contain the R-dimensional identity matrix, and whose remaining rows are filled with zeroes. However, we achieve rapid convergence by initializing the columns of $\boldsymbol{W}$ with the following set of orthonormal vectors, computed from the first R+7 frames of the image sequence using a Gram-Schmidt process [28]:

$$\begin{aligned} V_1 &= \sum_{t=1}^{8} X(t); \\ V_r &= X(r+7) - \sum_{s=1}^{r-1} \frac{X(r+7)^T V_s}{V_s^T V_s} V_s, \quad r = 2, \ldots, R; \\ W_r(R+7) &= V_r / \|V_r\|, \qquad\qquad r = 1, \ldots, R. \end{aligned} \tag{5}$$

Thus, the first basis vector is initialized as the normalized mean of the first eight frames, and each successive basis vector is initialized as the normalized residual between the next frame and its projection onto the subspace spanned by the previously-obtained basis vectors.

For real-time background suppression, the projection residuals are computed before

updating the basis vector estimates at each new frame. The temporal estimate of the background component in image frame $X(t)$ is computed by projecting $X(t)$ onto the space spanned by the columns of $\boldsymbol{W}(t\text{-}1)$. That is:

$$B_{TEMPORAL}(t) = \boldsymbol{W}(t-1)\boldsymbol{W}^{T}(t-1)X(t) \tag{6}$$

To determine whether the intensities observed at time t are consistent with the temporal model, the residual vector $X(t) - B_{TEMPORAL}(t)$ is normalized with appropriate pixelwise standard deviation estimates (Section 3.2) and each element is compared to the threshold, T_1.

To illustrate the effectiveness of subspace projection in jitter mitigation, a simple example of its application is provided in Figures 1 and 2. The analysis employs the 60×60 block of pixels that is highlighted in the full image of Figure 1. This block, which contains steep spatial gradients in various orientations, is used in the generation of a synthetic jitter sequence. The block is replicated 101 times, incorporating translational jitter via bicubic interpolation. The jitter displacements are independent and identically distributed (IID) bivariate Gaussian, with mean zero and standard deviation 0.25 pixels in the row and column dimensions. White Gaussian noise is added to each jittered version, with a standard deviation set to 0.01 of the standard deviation of the uncontaminated pixel intensities. That is, the clutter-to-noise ratio is equal to 100.

The first 10 frames are used to initialize $R = 3$ basis vector estimates as in (5), and FAPI is run through the remaining 90 frames, with decay rate β_1 set to 0.95. The FAPI-generated basis vector estimates at time $t = 100$ are shown in panels 2a, 2b, and 2c. The first basis vector is estimates the mean image, while the next two capture variability due to jitter in different directions. Three sets of background suppressed versions of the final image, $X(101)$, were computed using three different approaches. The residual surface obtained by subtracting the sample mean image is shown in panel 2d, the projection residual from the FAPI estimates is shown in panel 2e, and the projection residual from the first three eigenvectors of covariance matrix $\boldsymbol{C}_{XX}(100)$ is shown in panel 2f. All three sets of residuals are plotted on the same scale. The superiority of the projection approach is immediately apparent, and the similarity of the FAPI residuals to those obtained with a full covariance matrix decomposition is encouraging.

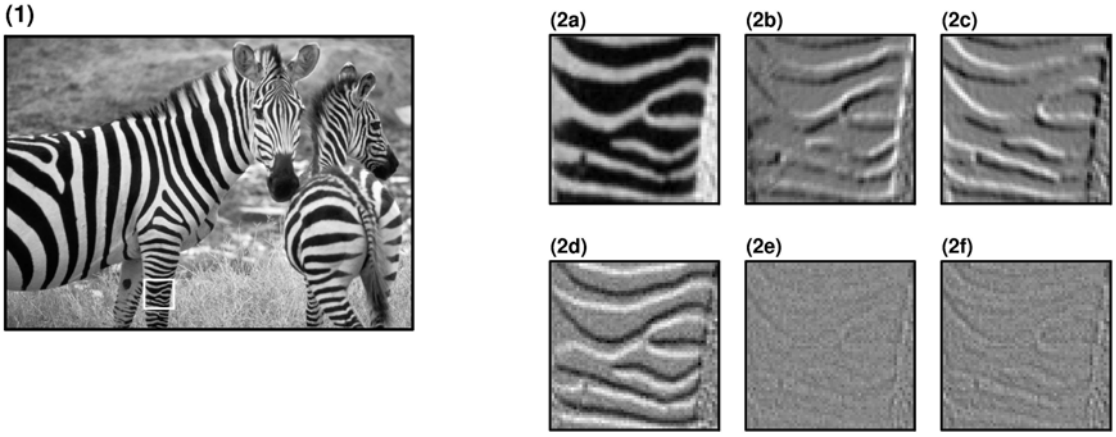

Figure 1 – Image exhibiting steep intensity gradients

This 620×860-pixel image shows steep gradients along many orientations. The 60×60-pixel area used to illustrate background suppression is highlighted.

Figure 2 – Basis vector decomposition

The three basis vectors computed using FAPI are shown in panels (2a-c). (2d) Residuals from subtracting the mean frame. (2e) Residuals from projecting into the FAPI subspace. (2f) Residuals from projecting into the subspace spanned by the first three eigenvectors of the 100-sample covariance matrix.

The choice of an appropriate value for the decay rate β_1 depends on the application at hand. Generally, we want to accommodate slow drift in the background, while preventing brief transient changes from being immediately incorporated into the basis vectors (and thus no longer detected). Suppose that a new object appears in the scene and then remains stationary, or that an indicator light turns on and stays that way. At some point, we may want to stop reporting "change" in the pixels containing the new object or event, so that it can be absorbed into the background model and a return to the previous state (object absent, indicator off) will be detected. To allow for application-specific tuning of the rate at which changes are incorporated into the basis vectors, we introduce a new parameter, β_2, to the basic FAPI updating formula. At time *t*, define the quantity $\tilde{x}(k,h;t)$ as follows:

$$\begin{aligned}\tilde{x}(k,h;t) &= \beta_2 b_{TEMPORAL}(k,h;t-1) + (1-\beta_2)x(k,h;t), \text{ if } \left|z_{MIN}(k,h;t)\right| > T_2 , \\ &= x(k,h;t), \text{ otherwise.}\end{aligned} \tag{7}$$

When computing the updated weight matrix $\boldsymbol{W}(t)$, the vector $\tilde{X}(t)$ with elements $\tilde{x}(k,h;t)$

is used in place of the observed data vector, $X(t)$. This directs the basis vector update away from pixel values with large normalized residuals. Like the decay rate β_1, parameter β_2 lies in the range [0,1]. The threshold T_2 may be set equal to the detection threshold T_1, or to another value appropriate to the application at hand. This feature is disabled by setting β_2 equal to zero.

3.2 – Temporal Variances

The temporal standard deviation estimate for pixel (k,h), based on data prior to time t, is denoted $\xi_{TEMPORAL}(k,h;t\text{-}1)$. The estimates are initialized from the same R+7 frames used to start the R basis vectors:

$$\xi^2_{TEMPORAL}\,(R+7) = \frac{1}{R+6}\sum_{u=1}^{R+7}\left[x(k,h;u) - b_{TEMPORAL}\,(k,h;R+7)\right]^2 . \qquad (8)$$

They are updated with each subsequent frame in the following manner:

$$\xi^2_{TEMPORAL}\,(k,h;t) = \gamma\xi^2_{TEMPORAL}\,(k,h;t-1) + (1-\gamma)\left[x(k,h;t) - b_{TEMPORAL}\,(k,h;t)\right]^2 . \qquad (9)$$

The variance forgetting factor, $\gamma \in [0,1]$, controls the update rate for each new observed residual. As with background estimation, we allow for the use of two different rates, with the slower rate selected for pixels with large normalized residuals. Specifically, we set $\gamma = \gamma_1$ in equation (9) if $z_{MIN}(k,h;t)$ lies below threshold T_2 and $\gamma = \gamma_2$ otherwise. Due to the strong influence that large squared residuals can have, we generally set γ_2 equal to 1.0, which prevents *any* updating of the temporal variance estimate for a pixel with large normalized residual.

With a sufficient training sample, temporal estimates will capture the behavior of pixels that are variable due to natural phenomena or sensor readout noise. As long as jitter is occurring while the temporal estimates are initialized and updated, this will also be reflected in larger estimated variances at pixels situated along high scene gradients. However, temporal variance estimates cannot anticipate the large pixel intensity differences that may be observed if the sensor platform is subject to sudden increases in the magnitude of the jitter. This is the circumstance for which the spatial models are designed.

4 – SPATIAL MOMENT ESTIMATION

While temporal moment estimates are computed from a time history of pixel intensities, the spatial estimates are computed from a single previous frame, or background estimate. The approach to computing spatial moments is based on the realization that one does not need to observe line-of-sight jitter over multiple frames to know which pixels may exhibit large intensity changes in the presence of such jitter. Pixels located in regions of high scene intensity gradient are obviously more susceptible than those lying in relatively homogeneous regions, and the variability of each pixel's intensity in the presence of jitter can be estimated from the pixel values in its immediate neighborhood. This is accomplished using a grid of conditional expectations in the vicinity of each pixel.

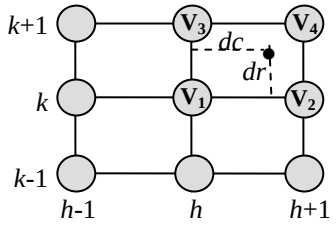

Figure 3 – Grid of pixel values

Consider the example shown in Figure 3. The nodes represent pixel centers in an arbitrary 3 × 3 sub-region of a larger image. At time t-1, the value at pixel (k, h) is equal to V_1, with values at nearby pixels given by V_2, V_3, and V_4. Suppose it is known that the jitter occurring between times t-1 and t will shift the sensor so that pixel (k, h) will be centered at the position corresponding to (k+dr, h+dc) at time t. In this case, bilinear interpolation could be used to predict the value observed at pixel (k, h) at time t in the following manner:

$$\hat{x}(k,h;t) = V_1 + dr(V_3 - V_1) + dc(V_2 - V_1) + (dr)(dc)(V_1 + V_4 - V_2 - V_3) \ . \qquad (10)$$

If the true shift (dr, dc) is unknown, as will generally be the case, its statistical distribution can be used to estimate the mean and variance of each pixel at time t as a function of the pixel values at time t-1. This is accomplished using a grid of conditional expectations in the neighborhood of each pixel location. We begin by assuming that the statistical distribution of the

jitter is known. For example, suppose that the row and column shifts occurring between times t-1 and t are IID Gaussian with mean zero and standard deviation σ. From this distribution, we can compute the probability of the sensor jittering in such a manner that pixel (k, h) will be centered in any of the "cells" in the vicinity of (k, h) at time t.

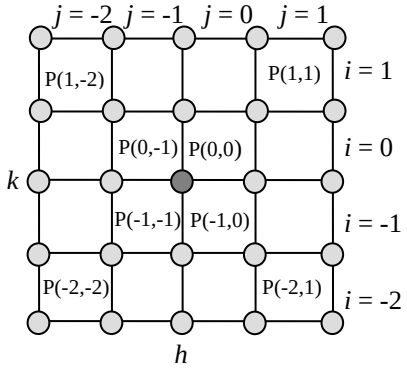

Figure 4 – Grid of cell probabilities
Cells in the vicinity of pixel (k, h) are shown.

Let $A_{ij}(t)$ represent the event that the jitter occurring between time t-1 and t has a row displacement between i and i+1 and a column displacement between j and j+1. The probability of event $A_{ij}(t)$ is denoted P(i, j). Note that the probability does not depend on the time t, as the jitter distribution is assumed (for now) to be stationary. Nor does P(i, j) depend on the location (k, h) of the particular pixel whose value we are predicting, as jitter is assumed to manifest itself as a rigid translation over the entire image. Thus, for any arbitrary pixel (k, h), the quantity P(i, j) represents the probability that jitter will re-center pixel (k, h) to a row position between k+i and k+i+1, and a column position between h+j and h+j+1. The grid of probabilities in the vicinity of pixel (k, h) is depicted in Figure 4.

The actual probabilities can be computed in a straightforward manner from the assumed bivariate Gaussian distribution of row and column displacements. In particular:

$$\mathrm{P}(i,\,j) = \left[\Phi\left(\frac{i+1}{\sigma}\right) - \Phi\left(\frac{i}{\sigma}\right)\right]\left[\Phi\left(\frac{j+1}{\sigma}\right) - \Phi\left(\frac{j}{\sigma}\right)\right], \qquad (11)$$

where Φ represents the probability distribution function of a standard (zero mean, unit variance) Gaussian random variable. Probabilities computed using (11) will be concentrated in a small block of cells centered at node (*k*, *h*). For any value of σ specified, a box size can be calculated such that the probability of jittering to a location within a *BOX* × *BOX* region centered on (*k*, *h*) exceeds 99%. To find the appropriate box size, compute the quantity *Y* as follows:

$$Y = -\sigma\,\Phi^{-1}\left(\frac{1-\sqrt{0.99}}{2}\right) = 2.806\sigma \ . \tag{12}$$

Here, Φ^{-1} is the inverse standard Gaussian distribution function. Numerical evaluation gives 2.806 as the Gaussian quantile corresponding to probability $\left(1-\sqrt{0.99}\right)/2$. It follows that the minimum box size required to ensure a 99% coverage probability has dimension equal to twice the smallest integer that exceeds $2.806\,\sigma$. Thus, if σ = 1/4 pixel, we see that *Y*= 0.702 and a 2 × 2 cell region centered on (*k*, *h*) has at least a 99% chance of containing the position jittered to at time *t*. For σ = 1/2 pixel, a 4 × 4 cell region like the one depicted in Figure 4 has at least a 99% coverage probability, while a 6 × 6 cell region will suffice for σ = 1. Cells lying outside of the centered region calculated for a given value of σ have very low probability and thus minimal impact on the spatial variance estimates. Omitting such outlying cells from the calculations that follow provides a substantial improvement in computational run-time.

Once an appropriate box size has been identified, expression (11) is used to compute the probability of each cell within the box. These probabilities are normalized by the total probability of the centered *BOX* × *BOX* region. The normalizing factor, P_{TOT}, is calculated by inverting (12):

$$P_{TOT} = \left[1 - 2\Phi\left(\frac{BOX}{2\sigma}\right)\right]^2 \ . \tag{13}$$

The normalized probability estimate for the cell bounded by rows *k*+*i* and *k*+*i*+1, and columns *h*+*j* and *h*+*j*+1 is denoted $\tilde{P}(i, j)$ and is given by:

$$\tilde{\mathrm{P}}(i,\, j) = \mathrm{P}(i,\, j) / \mathrm{P}_{\mathrm{TOT}} \,. \tag{14}$$

Normalization ensures that the sum of the cell probabilities equals unity. Because $\mathrm{P}_{\mathrm{TOT}}$ is defined to lie between 0.99 and 1.0, the adjustments made in (14) are always minor.

The next step in spatial moment estimation is calculating the conditional expected value of pixel (*k*, *h*) at time *t*, given the that its center has jittered into a specific cell, and assuming the bilinear model (10). This quantity depends on the exact row and column shifts (*dr* and *dc*) within the cell to which the pixel has jittered, and the values of the four pixels bordering the cell at time *t*-1. Of these six quantities, only *dr* and *dc* are unknown at time *t*-1. The intensity estimate (10) is a simple function of the two unknowns, so its mean and mean-square can be computed in a straightforward manner. Begin by defining the following quantities for algebraic convenience:

$$\begin{aligned} \mathrm{D}_1 &= \mathrm{V}_1 \\ \mathrm{D}_2 &= \mathrm{V}_3 - \mathrm{V}_1 \\ \mathrm{D}_3 &= \mathrm{V}_2 - \mathrm{V}_1 \\ \mathrm{D}_4 &= \mathrm{V}_1 + \mathrm{V}_4 - \mathrm{V}_2 - \mathrm{V}_3 \ . \end{aligned} \tag{15}$$

As in Figure 3, V_1 represents the pixel value at time *t*-1 in the lower left-hand corner of the cell into which jitter has occurred, while V_2, V_3, and V_4 are the lower right, upper left and upper right corners, respectively. All of these quantities are known at time *t*-1. From (10), it follows that the conditional expected values are given by:

$$\begin{aligned} \mathrm{E}[\hat{x}(k,h;t) \mid A_{ij}(t)] = \mathrm{D}_1 + \mathrm{D}_2\, \mathrm{E}[dr \mid A_{ij}(t)] + \mathrm{D}_3\, \mathrm{E}[dc \mid A_{ij}(t)] + \\ \mathrm{D}_4\, \mathrm{E}[dr \mid A_{ij}(t)]\mathrm{E}[dc \mid A_{ij}(t)] \end{aligned} \tag{16a}$$

$$\begin{aligned} \mathrm{E}[\hat{x}^2(k,h;t) \mid A_{ij}(t)] = \mathrm{D}_1^2 + 2\,\mathrm{D}_1\mathrm{D}_2\mathrm{E}[dr \mid A_{ij}(t)] + 2\,\mathrm{D}_1\mathrm{D}_3\mathrm{E}[dc \mid A_{ij}(t)] + \\ 2(\mathrm{D}_1\mathrm{D}_4 + \mathrm{D}_2\mathrm{D}_3)\,\mathrm{E}[dr \mid A_{ij}(t)]\mathrm{E}[dc \mid A_{ij}(t)] + \mathrm{D}_2^2\mathrm{E}[dr^2 \mid A_{ij}(t)] + \\ \mathrm{D}_3^2\mathrm{E}[dc^2 \mid A_{ij}(t)] + 2\,\mathrm{D}_2\mathrm{D}_4\mathrm{E}[dr^2 \mid A_{ij}(t)]\mathrm{E}[dc \mid A_{ij}(t)] + \\ 2\,\mathrm{D}_3\mathrm{D}_4\mathrm{E}[dr \mid A_{ij}(t)]\mathrm{E}[dc^2 \mid A_{ij}(t)] + \mathrm{D}_4^2\mathrm{E}[dr^2 \mid A_{ij}(t)]\mathrm{E}[dc^2 \mid A_{ij}(t)] \ . \end{aligned} \tag{16b}$$

We see in (16a) and (16b) that the mean and variance of the pixel value estimates obtained using bilinear interpolation are functions of the conditional expected values of the row and column displacements and their squares, given jitter into a specific cell. These conditional expectations are calculated using the truncated Gaussian distribution [13]. In particular:

$$\mathrm{E}[dr \mid A_{ij}(t)] = \left[\frac{\phi\left(\frac{i}{\sigma}\right) - \phi\left(\frac{i+1}{\sigma}\right)}{\Phi\left(\frac{i+1}{\sigma}\right) - \Phi\left(\frac{i}{\sigma}\right)} \right] \sigma - i \ , \tag{17a}$$

$$\mathrm{E}[dr^2 \mid A_{ij}(t)] = \sigma^2 + i^2 - \frac{i\sigma\,\phi\left(\frac{i}{\sigma}\right) - (i-1)\sigma\,\phi\left(\frac{i+1}{\sigma}\right)}{\Phi\left(\frac{i+1}{\sigma}\right) - \Phi\left(\frac{i}{\sigma}\right)} \ . \tag{17b}$$

Note that ϕ represents the density function of the standard Gaussian distribution. Conditional expectations for the column displacements $\mathrm{E}[dc|A_{ij}(t)]$ and their squares are calculated for each column j in the same manner.

Substituting the quantities (17a) and (17b), and their equivalents for column displacement, into (16a) and (16b) gives expressions for the expected value and mean-square in pixel (k, h) at time t, given jitter into the cell bounded by rows $k+i$ and $k+i+1$, and columns $h+j$ and $h+j+1$. The Law of Total Probability is then invoked to estimate the *unconditional* expectations:

$$b_{SPATIAL}(k,h;t) \equiv \mathrm{E}[\hat{x}(k,h;t)] = \sum_i \sum_j \mathrm{E}[\hat{x}(k,h;t) \mid A_{ij}(t)] \cdot \tilde{\mathrm{P}}(i,j) \tag{18a}$$

$$\mathrm{E}[\hat{x}^2(k,h;t)] = \sum_i \sum_j \mathrm{E}[\hat{x}^2(k,h;t) \mid A_{ij}(t)] \cdot \tilde{\mathrm{P}}(i,j) \, . \tag{18b}$$

The double sums in (18) each run from a lower limit of $-BOX/2$ to an upper limit of $BOX/2 - 1$. Finally, the spatial estimate of the variance of pixel (k, h) at time t, denoted $\xi^2_{SPATIAL}(k,h;t)$, is

calculated from first principles:

$$\xi^2_{SPATIAL}(k,h;t) = \mathrm{var}[\hat{x}(k,h;t)] = \mathrm{E}[\hat{x}^2(k,h;t)] - \mathrm{E}^2[\hat{x}(k,h;t)]. \quad (19)$$

With this approach, spatial moment estimates for every pixel (*k*, *h*) and time *t* are computed based only on the pixel values in frame *t*-1 and the assumed jitter model. An example of a spatial standard deviation map is depicted in Figure 5. The 60 × 60 block highlighted in Figure 1 is shown in panel 5a, while panel 5b is the spatial standard deviation map computed for jitter standard deviation σ = 0.5 pixel.

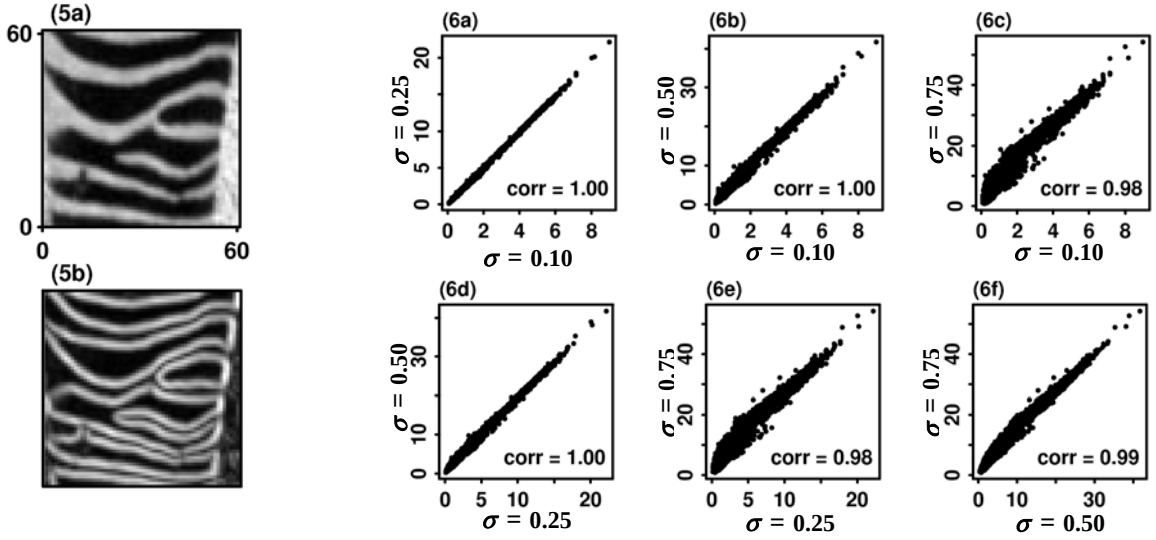

Figure 5 – Image block and spatial standard deviation map
(5a) 60 × 60 block of pixels. (5b) Spatial standard deviation map for σ = 0.50 pixel.

Figure 6 – Effect of jitter standard deviation setting
Scatterplots show the relationship between spatial standard deviations calculated with different values of the jitter standard deviation. (a) σ = 0.10 vs. σ = 0.25. (b) σ = 0.10 vs. σ = 0.50. (c) σ = 0.10 vs. σ = 0.75. (b) σ = 0.25 vs. σ = 0.50. (b) σ = 0.25 vs. σ = 0.75. (b) σ = 0.50 vs. σ = 0.75.

Spatial moment estimates are surprisingly robust to misspecification of the jitter standard deviation. For example, experience has shown that two sets of spatial standard deviation estimates computed using (11) - (19), but using different values of σ, differ approximately by a constant of proportionality, as long as the jitter standard deviations are of the same order of magnitude. The relationship can be seen in Figure 6, which shows pairwise comparisons between spatial standard deviations computed with σ = 0.10, 0.25, 0.50, and 0.75 pixels. In all cases, the

correlation coefficient exceeds 0.95, indicating a strong linear relationship between the various sets of estimates. This leads us to develop a strategy for calculating frame-specific scale factors that can be applied to compensate for the actual level of jitter that has occurred.

The reasoning is as follows. Suppose that we have computed spatial standard deviations $\xi_{SPATIAL}(k,h;t)$ for each pixel (k, h) at time t, using the frame observed at time t-1 and $\sigma = 0.5$ pixels. Suppose further that the actual jitter occurring between times t-1 and t results in a row displacement of 0.01 pixel and a column displacement of 0.05 pixel, far lower than expected under the model. If jitter is the primary source of pixel intensity differences, we expect that raw spatial differences normalized by the spatial standard deviations $\xi_{SPATIAL}(k,h;t)$ will tend to be small in absolute value, particularly along scene gradients. Conversely, if the displacement at time t is unusually large, normalized differences will tend to be large in absolute value along the scene gradients. To compensate for this effect, we re-scale the spatial standard deviation estimates on a frame-by-frame basis, using a scale factor that is calculated exclusively from pixels that show relatively strong spatial gradients, and do not have unusually large initial residuals. This is achieved as follows:

1. Compute initial normalized differences $z_{SPATIAL}(k,h;t)$ as in (1b).

2. Identify candidate outlier pixels as in Tukey [27]: Sort the normalized residuals, and define the interquartile range (IQR) to be the difference between the 75th percentile and the 25th percentile of the values. Pixels with normalized residuals 1.5 × IQR lower than the 25th percentile or 1.5 × IQR higher than the 75th percentile are considered outliers. These are potential scene changes, which should be excluded from jitter normalization.

3. Identify the pixels that are most likely to be dominated by jitter noise: those with strong spatial gradients. Define the set Ω, of size N_Ω, to contain all non-outlier pixels (k,h) such that $\xi_{SPATIAL}(k,h;t)$ is above the 80% percentile of all spatial standard deviation estimates at time t. Over the set Ω, calculate the following:

$$m(t) = \frac{1}{N_\Omega}\sum_{\Omega} z_{SPATIAL}(k,h;t)\,, \quad \mathrm{S}^2(t) = \frac{1}{(N_\Omega - 1)}\sum_{\Omega}[z_{SPATIAL}(k,h;t) - m(k,h;t)]^2 \qquad (21)$$

4. Finally, multiply the spatial standard deviation estimate at every pixel by $S(t)$:

$$\tilde{\xi}_{SPATIAL}(k,h;t) = S(t)\xi_{SPATIAL}(k,h;t)\ . \qquad (22)$$

Note that the upper and lower outlier thresholds (step 2) and the 80th percentile of the spatial standard deviations (step 3) are unlikely to undergo large changes quickly. Thus, for real-time applications they do need not be updated with every frame. The scale factor $S(t)$ compensates for differences between the actual jitter displacement at time t, and the model specified in computing spatial standard deviations. This is achieved *without* direct estimation of the actual jitter displacements (i.e., frame registration).

To keep target changes from contaminating the spatial moment estimates at time t, the moments can be calculated using an estimated background frame prior to time t, rather than a single prior frame (which may contain targets). In particular, we can obtain the pixel values V_1, V_2, V_3, and V_4 for use in (10) – (18) from a temporal background estimate $B_{TEMPORAL}(t - s)$ computed as in Equation (6) from a prior frame, $X(t - s)$, where $s \geq 1$. If the sensor is subject to drift in pointing or focus, it may be necessary to update spatial moments with every frame. If the sensor, and thus the background, is expected to remain fairly stable aside from jitter effects, then a lower update rate may be preferable. Examples illustrating both cases are included in Section 6.

5 – IMPLEMENTATION

The ten parameters used to tune the change detection algorithm are summarized in Table 1. All have been introduced previously with the exception of T_3, which is employed to prevent false alarms due to very small intensity changes in low-variance pixels. T_3 is applied as follows. For pixel (k, h) at time t, if the numerators of (1a) and (1b) (the raw temporal and spatial deviations about background) each lie below T_3 in absolute value, then neither detection nor background suppression occurs, regardless of the magnitudes of the normalized differences.

Table 1 – Parameters of the change detection algorithm

Parameter:	Description:
R	Number of basis vectors used in temporal background estimation.
β_1	Decay rate applied in FAPI basis vector updates.
β_2	Decay rate for target suppression in FAPI basis vector updates.
γ_1	Default forgetting factor for temporal variance estimation.
γ_2	Forgetting factor for temporal variance estimation, with target suppression.
σ	Jitter standard deviation for spatial moment estimation.
T_1	Minimum normalized difference threshold for target detection.
T_2	Minimum normalized difference threshold for target suppression.
T_3	Minimum raw difference threshold for target detection.
NEIGH	Minimum number of pixels in a contiguous cluster of detections.

The number of basis vectors, *R*, is set to 3 in all of our applications. Selecting a larger number sometimes produces cleaner looking residual surfaces. However, experience has shown that detector performance may suffer due to an increase in false alarms in unstructured parts of the scene, which may not be well represented in a model subspace that is dominated by strong spatial gradients. The decay parameter β_1 should be large enough to prevent small transient changes from being immediately incorporated into the background estimate, but small enough to track gradual change. Values of β_1 between 0.90 and 0.99 are suitable for most problem types. Suppression of target changes from the background subspace is accomplished using parameter β_2, which we generally set at or above 0.99, depending on whether we wish to allow gradual incorporation of persistent change. The forgetting factors γ_1 and γ_2 control the rate at which temporal variance estimates are updated with each new frame. We recommend setting γ_1 = 0.99 as a default, with γ_2 fixed at 1.0 for robustness to real change or anomalous readings.

The remaining five parameters (jitter standard deviation σ, thresholds T_1, T_2, and T_3, and number of connected detection pixels *NEIGH*) are application-specific. Setting σ conservatively, based on the worst jitter anticipated, is a reasonable strategy in most cases. Appropriate values of thresholds T_1, T_2, and T_3 will depend on the characteristics of the noise environment and the expected strength of the target change signals. Finally, *NEIGH* may be adjusted to the size (spatial extent) of the changes to be detected. In the examples that follow, which represent two very different change detection scenarios, the first five parameters are held fixed throughout, while the final five are tuned to the specific data sets.

6 – ILLUSTRATIVE EXAMPLES

In this section, the application of our change detection algorithm is illustrated in two distinct problem domains. The first case uses a natural scene with an extremely dynamic background, and target changes that are small and rare. By contrast, the second scenario involves a highly structured, man-made scene, with target changes that are large, frequent, and persistent.

6.1 – Natural Scene

Several frames of a video sequence downloaded from the on-line Bear Camera at the Woodland Park Zoo in Seattle are shown in Figure 7. To provide scene context, panel 7a is displayed in color, although all of our detection work is conducted in single-band greyscale, as seen in panels 7b and 7c. To appreciate the challenge associated with this dynamic scene, the interested reader is encouraged to view live video via the Zoo's website [29]. A shallow stream runs through the middle of scene, then flows over a log and into a pool at the bottom of the frame. The intensity of the stream pixels is highly variable, due to water motion, along with surface shimmer in the pool. The resolution and focus of the camera are moderate, and the platform is stable, with little observable jitter.

To test our change detection algorithms, we induced jitter of varying magnitudes on a sequence of 300 frames, sampled from the video at 10 Hz. The data were first converted to greyscale, and then each frame was randomly jittered according to a specific distribution (discussed shortly) using bicubic interpolation. The jittered frames were cropped to 200 × 380 pixels, both to remove any edge artifacts from the interpolation and to prevent false detections on changes in the digital time stamp. Two jittered, truncated frames are shown in panels 7b and 7c.

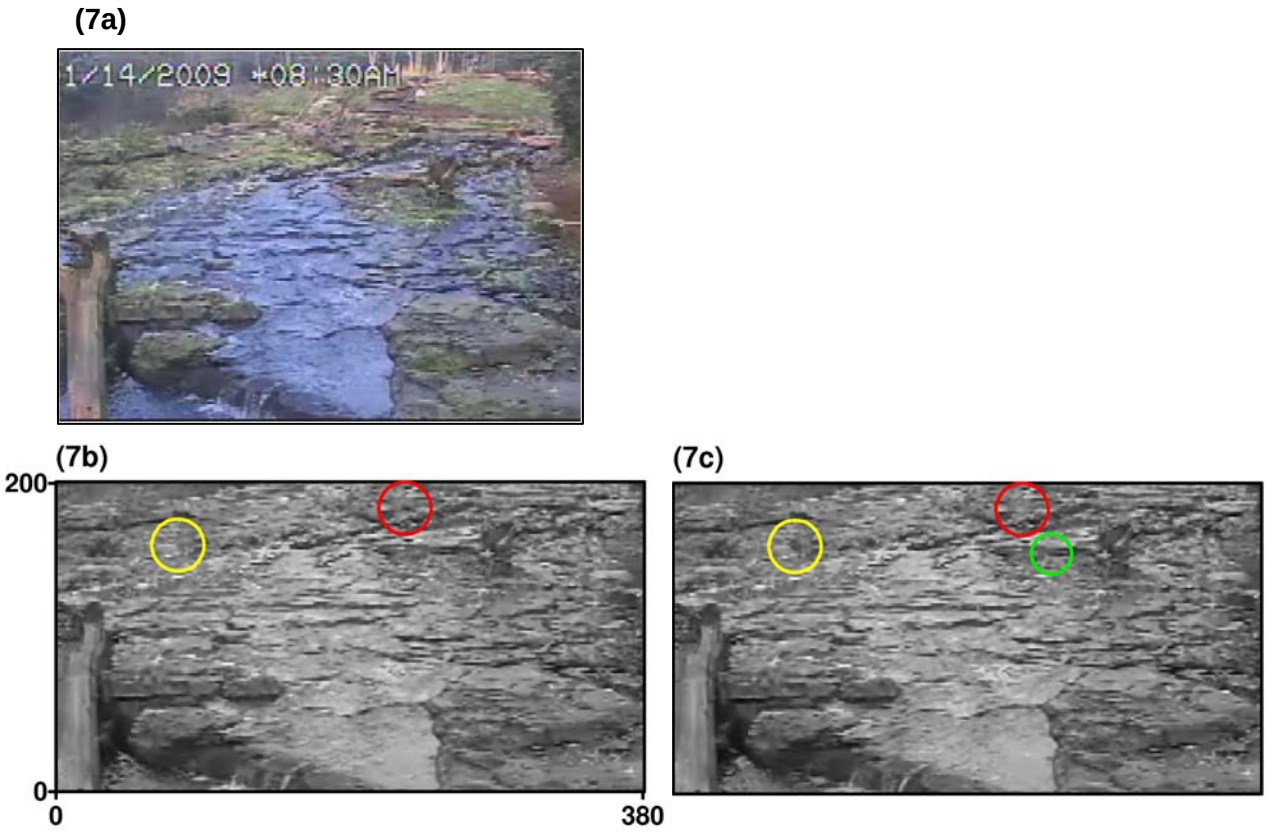

Figure 7 – Bearcam sequence
(a) Frame 1 of the Bearcam 0 sequence, shown in color to provide context. (b) A small black bird (red circle) flies into the scene at frame 213. (c) In frame 223, the bird (yellow circle) lands at the edge of the stream. It is difficult to pick out these changes without proper background suppression. The pixel whose history is plotted in Figure 8 lies at the center of the green circle.

The original sequence is referred to as Bearcam 0, and three separate jittered frame sequences were generated from it, as summarized in Table 2. The first jittered sequence, Bearcam 1, used a bivariate Gaussian distribution, with independent row and column translations each having mean $\mu = 0$ and standard deviation $\tau = 0.50$ pixels. The sequence is stationary, meaning that each frame is offset from its own initial position, rather than the shift applied to the previous frame. The Bearcam 2 sequence is also stationary, but here row and column translations were generated from a 5% Wild, or "scale contaminated Gaussian" distribution [9]. At each frame, the row and column shifts had a 95% chance of being drawn from a zero-mean Gaussian with $\tau = 0.25$, and a 5% chance of coming from a zero-mean Gaussian with $\tau = 1.0$. The final sequence, Bearcam 3, is non-stationary. Incremental row and column shifts were independently generated for each frame from a zero-mean Gaussian with $\tau = 0.25$, and then added to the cumulative sum of all previous shifts. This allowed for a gradual drift in pointing, as may occur

with a sensor placed on a slowly moving platform. Over the 300 frames that we analyzed, the row and column positions for the non-stationary sequence drifted by 4.4 and 8.9 pixels, respectively.

Table 2 – Jitter characteristics for the Bearcam sequences

Sequence	Jitter Distribution
Bearcam 0	Control sequence, no induced jitter
Bearcam 1	Bivariate IID Gaussian: mean $\mu = 0$, stdev $\tau = 0.50$; stationary
Bearcam 2	Bivariate IID 5% Wild: 95% $\mu = 0$, $\tau = 0.25$; stationary 5% $\mu = 0$, $\tau = 2.5$; stationary
Bearcam 3	Bivariate IID Gaussian: $\mu = 0$, $\tau = 0.25$; non-stationary

Change detection for the Bearcam sequences proceeded in the following manner. The first ten frames were used to initialize three basis vectors as in (5), and pixelwise temporal variances as in (8). An additional 90 frames were then run through FAPI, with basis vector and temporal variance estimates updated at each iteration. Detection was not run on these first 100 frames, so candidate targets were not suppressed from the moment estimates; this is equivalent to setting the parameters $\beta_2 = 0$ and $\gamma_2 = \gamma_1$ during the training period. The initial spatial moment estimates were generated from FAPI background frame $B(100)$, and full detection (with target suppression) commenced with frame 101. Spatial moment estimates were re-calculated for every frame, using pixel values from the temporal background estimate at the immediate previous frame. The parameter settings, identical for each test run, are listed in Table 3.

Table 3 – Parameter settings for the Bearcam and Sidewalk sequences

PARAMETER:	Bearcam Settings	Sidewalk Settings
R	3	3
β_1, β_2	0.975, 0.999	0.975, 0.999
γ_1, γ_2	0.99, 1.0	0.99, 1.0
σ	0.50	4.0
T_1, T_2, T_3	8.0, 8.0, 10.0	4.0, 3.0, 10.0
NEIGH	3	8

To illustrate the effectiveness of subspace projection in suppressing jitter effects, the temporal history of a single pixel through 100 frames of each Bearcam sequence is shown in Figure 8. The pixel's location (circled in panel 7c) lies near a significant spatial gradient. Accordingly, while its intensity is nearly constant in the control sequence, it varies considerably in each of the jittered sequences. When the jitter distribution is stationary, the FAPI subspace does an excellent job of modeling background, resulting in small projection residuals at each frame. For the drifting sequence, Bearcam 3, reducing decay parameter β_1 from 0.975 to 0.925 allows FAPI to better incorporate gradual scene change into the background subspace.

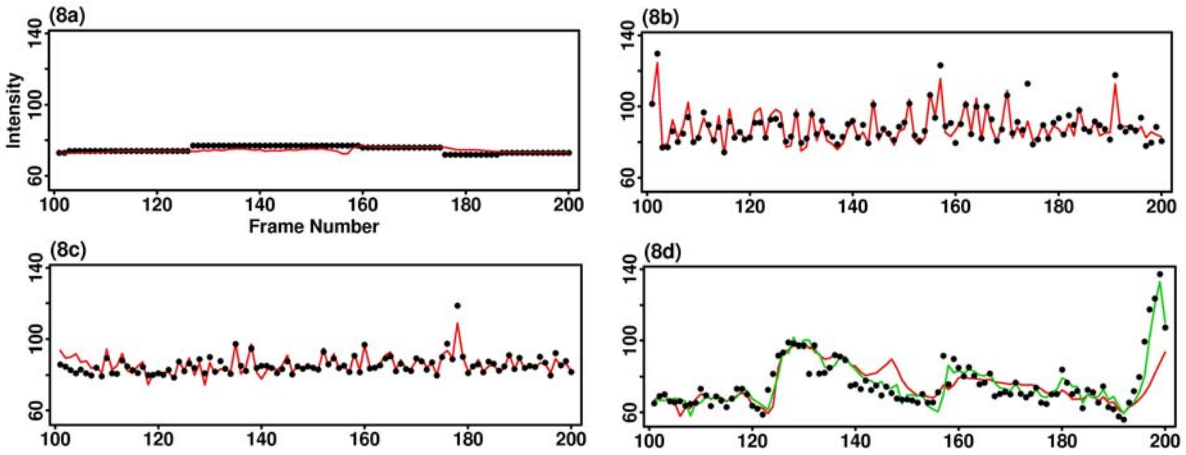

Figure 8 – A single pixel's history through 100 frames
The history of pixel (155, 245) through 100 frames in each of the four Bearcam sequences is shown. The location of this pixel is highlighted in Figure 7. (a) Control sequence, Bearcam 0. The red line is the running background estimate obtained by subspace projection. (b) Sequence Bearcam 1. (c) Sequence Bearcam 2. (d) Sequence Bearcam 3. Red line: β1=0.975. Green line: β1=0.925.

The "target" change for detector performance analysis is a small black bird that flies into the scene in frame number 213 (panel 7b), lands on the ground in frame 223 (panel 7c), and stays in the vicinity of this position for the remainder of the frames. For each sequence, detector performance on frames 201 – 300 is measured by counting the number of frames (the maximum possible is 88) in which the bird is detected, along with the number of frames in which at least one false alarm (detected change not due to the bird) occurs. Results are summarized in Table 4.

Table 4 – Detection results for the Bearcam sequences

Sequence:	Target Detection Frames (of 88)	False Alarm Frames (of 100)
Bearcam 0	84	5
Bearcam 1	20	2
Bearcam 2	20	4
Bearcam 3	7	7

As should be expected, performance was best in the control case, Bearcam 0. Here, the bird was detected in 84 of the 88 frames in which it was present. Target detection dropped for the jittered sequences, largely due to the bird fading into the noisy background after remaining in place for an extended time. The last detection of the bird on any of the jittered sequences occurred in frame number 247, 24 frames after landing. In the control sequence, detection continued to the final frame, number 300.

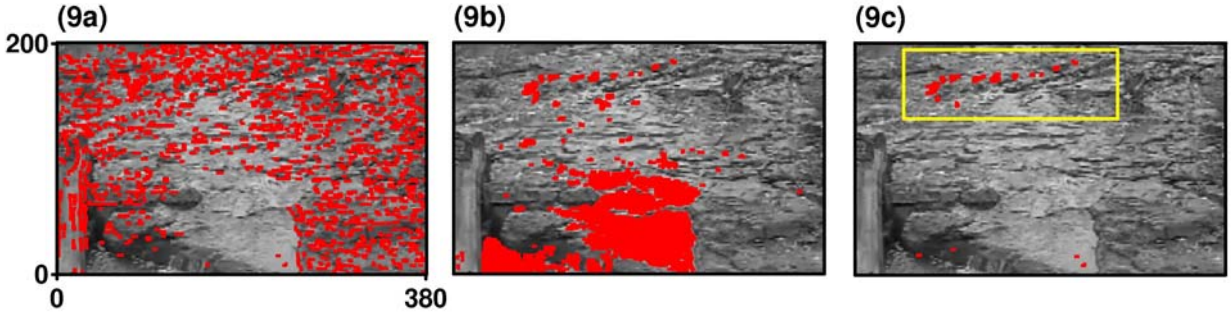

Figure 9 – Maps of threshold exceedance for the Bearcam 2 sequence
Combined results from frames 201 – 300 are shown. Pixels shown in red had normalized differences with absolute value above the detection threshold T1 for at least one of these 100 frames. (a) Temporal exceedances. (b) Spatial exceedances. (c) Pixels flagged as detections using the minimum ratio test of Equation (2). Detections within the yellow box are due to the target bird.

Maps showing threshold exceedances for the 5% Wild sequence, Bearcam 2, are shown in Figure 9. Pixel locations where the absolute value of the normalized temporal difference (1a) exceeded the detection threshold of T_1 in at least one frame of the 100-frame test sequence are highlighted in panel 9a. Pixels with absolute normalized spatial difference (1b) above T_1 in at least one frame are shown in panel 9b, and those with minimum absolute normalized difference (2) above the threshold are highlighted in panel 9c. Detections within the yellow box of panel 9c

are due to the bird or its shadow, and are considered true change signals. The remaining four detections are single-frame false alarms occurring in high-variance water pixels.

The advantage of the dual-model approach to handling jitter in dynamic scenes is obvious. Sudden changes in jitter magnitude may not map well into a subspace that was generated from previous low-jitter observations, resulting in large temporal deviations. This is evident in panel 9a, where many exceedances are observed, generally lying along scene gradients. By contrast, the purely spatial approach cannot adequately model pixels whose primary source of variability is not jitter-induced. This is seen in panel 9b, showing a large number of spatial exceedances on moving water pixels. Selecting the minimum of the two normalized ratios as a final measure of consistency allows us to take full advantage of benefits of each model. The combination of the two provides a powerful capability to discriminate true change.

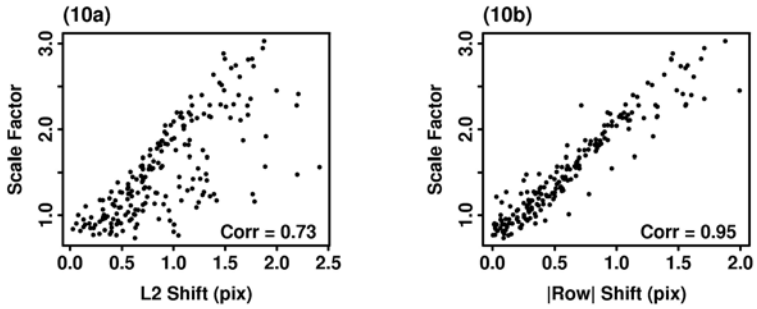

Figure 10 – Jitter displacements and spatial standard deviation estimates
The relationship between estimated spatial standard deviation scale factors and actual jitter displacements is shown, for 200 test frames in the Bearcam 1 sequence.

Recall that spatial standard deviation estimates are scaled (Equation 22) on a per-frame basis. Our assertion that the scale factors compensate for jitter without requiring knowledge or direct estimation of the actual displacements is well supported by Figure 10. For the test frames of the Bearcam 1 sequence, panel 10a shows the estimated scale factor as function of the L_2 distance between the true displacement in the frame used to estimate spatial moments (t-1), and the tested frame (t). Panel 10b shows the scale factor as a function of the absolute row displacement only. Because most gradients in the Bearcam scene are horizontal, the scale factors are more highly correlated with the absolute row displacements than with the L_2 distances. Even for this dynamic scene, it is clear that the scale factors are largely determined by the (simulated) jitter.

When the fence and 80th percentile values (Steps 1-2, Section 4) used in computing scale factors were updated with every frame, a processing rate of 13 frames per second (FPS) was achieved for the Bearcam sequences on a Sun server with eight dual-core 2.15 GHz processors. When fence and percentile values were computed for the first test frame only, a rate of 30 FPS was achieved on the same hardware, with no meaningful difference in detection performance.

6.2 - Man-Made Scene

The second example relies on 300 frames from a video sequence first published by Jodoin et al. [11]. The "Sidewalk" scene, shown in Figure 11, is highly structured with strong gradients in various orientations. The camera used to collect the frames has good focus and resolution, but is located close to ventilation fans on the building to which it is mounted. Motion of the fans induces substantial jitter at the camera's position, resulting in multiple-pixel displacements in the video stream. Numerous pedestrians move across the scene during the time period covering the training and test frames. These people and their shadows will be considered "target" changes. There is little natural variability in the area, so that most pixel intensity changes are caused by jitter, pedestrian motion, or camera noise. Parameter settings for the Sidewalk sequence are listed in Table 3. Due to the low noise level of the Sidewalk environment, the detection and suppression thresholds T_1 and T_2 are smaller than was appropriate for the noisier Bearcam sequences. The target changes occupy a large number of pixels, allowing us to increase the value of parameter *NEIGH* to 8 pixels. Finally, the camera displacements are substantial, requiring the increase of jitter standard deviation σ to 4.0 pixels for generation of spatial moments.

Because no target-free frames are available from which to start off the temporal and spatial moment estimates, a modified initialization scheme was developed based on [11]. The background subspace basis vectors were initialized from the first 10 frames as in (5). For frames 11 – 50, FAPI was used to update the basis vectors, without detection or target suppression enabled. Denote by $B_{TEMPORAL}(50)$ the background estimate following frame 50. The starting temporal variance estimates were computed as the *median* squared deviations about $B_{TEMPORAL}(50)$, computed over the frames 1 – 50. While several pedestrians are present in the scene during this training period, few (if any) pixels are occupied for more than half of the

frames. Thus, the median squared differences are fairly robust to the intermittent presence of targets. The initial spatial moment estimates are computed directly from $B_{TEMPORAL}(50)$. Because target suppression was not applied for the first 50 frames, $B_{TEMPORAL}(50)$ does contain some target energy from two pedestrians, as seen in Figure 11, panel 11b. To improve on the initial spatial estimates, training continues for another 50 frames, *with target suppression*. For frames 51 through 100, the temporal moment estimates were updated with every frame, but the spatial moments were held constant. Pixels whose minimum normalized differences exceeded the threshold T_2 were suppressed from the temporal background and standard deviation updates, via fade factors β_2 and γ_2, respectively.

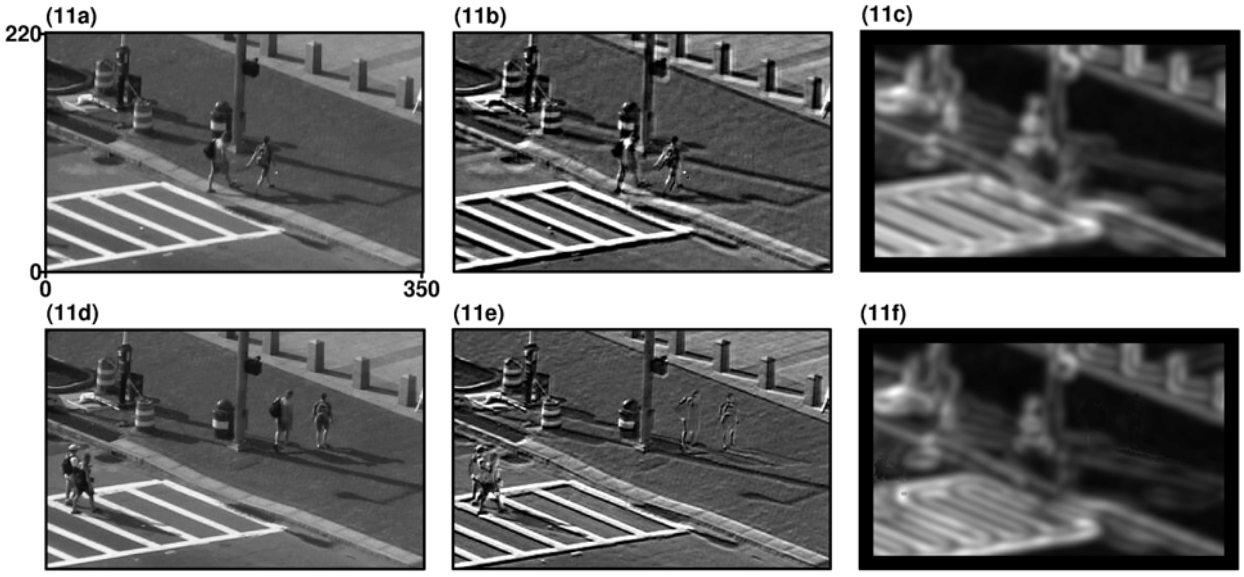

Figure 11 – Sidewalk sequence
Camera frames, FAPI background estimates, and spatial standard deviation maps are shown. (a) Frame 50. (b) Background estimate $\beta_{TEMPORAL}(50)$, computed without target suppression and showing significant jitter-induced structure, along with two pedestrians. (c) Initial spatial standard deviation map computed from $\beta_{TEMPORAL}(50)$. (d) Frame 100. (e) Background estimate $\beta_{TEMPORAL}(100)$. (f) Final spatial standard deviation map, computed from $\beta_{TEMPORAL}(100)$.

Frame 100 is shown in panel 11d, with associated background estimate $B_{TEMPORAL}(100)$ in panel 11e. Target suppression eliminates most pedestrian-induced structure in the background estimate for pixels on the relatively uniform sidewalk. However, target pixels overlapping with the crosswalk are neither detected nor suppressed, due to the large temporal and spatial standard deviations in this highly structured part of the scene. As a result, some target energy, mostly from the legs of two pedestrians, does make it into the final spatial moment estimates (panel

11f). Note that spatial moments are unavailable in a 12-pixel band around the edge of the frames, due to the large size of the box over which the moments are calculated.

The computational cost of updating spatial moments in a very high jitter environment is prohibitive. Setting $\sigma = 4.0$ pixels for the Sidewalk sequence results in a box size (Equation 12) of 24 × 24 pixels. Accordingly, the spatial moment estimates are held fixed at the values calculated following frame 100 for the remainder of the frame sequence, while temporal estimates are updated after every frame, with target suppression. In the case of a gradually drifting camera, a modified scheme could be implemented with spatial updates every K frames, where integer $K > 1$ is chosen to enable acceptable run-time performance.

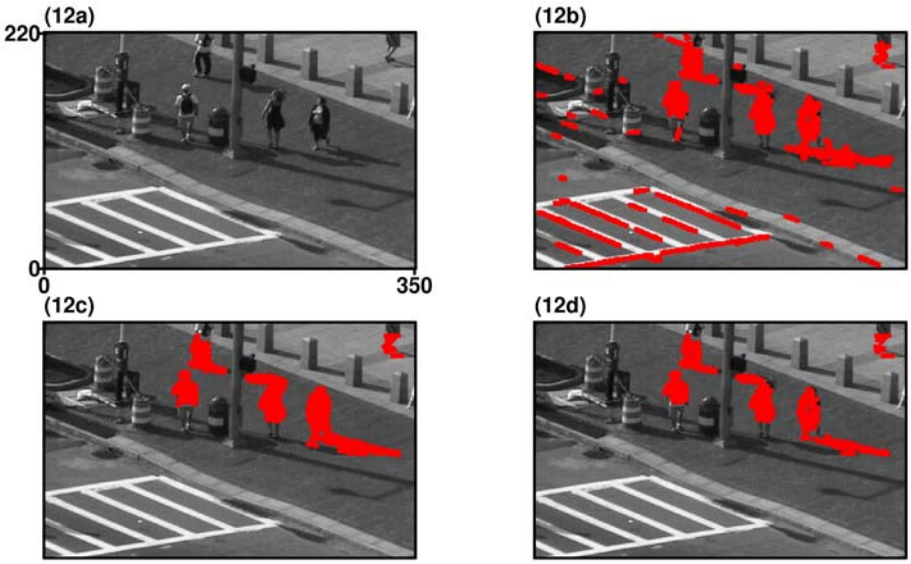

Figure 12 – Detection results for Sidewalk frame 228
(a) Camera frame 228. (b) Threshold exceedence map for normalized temporal differences. (c) Threshold exceedence map for normalized spatial differences. (d) Change detection map, based on minimum absolute normalized differences. Targets cannot be detected within 12 pixels of the edge of the frame, due to the unavailability of spatial moments.

The background subspace computed for the Sidewalk sequence successfully captures most of the effects of camera jitter on the frame data. For the vast majority of frames, the absolute normalized temporal differences were below detection threshold $T_1 = 4.0$ for all non-target pixels. However, unusually large jitter displacements did sometimes occur, for which spatial moments were required to mitigate false alarms. One such example, frame 228, is shown in Figure 12. Here, temporal false alarms occurred at many pixels lying along scene edges. These

jitter-induced exceedances were all rejected by the spatial model, which properly down-weighted pixels whose intensities were most susceptible to jitter effects. In the change detection map of panel 12d, all five pedestrians are readily detected, and there are no false alarms.

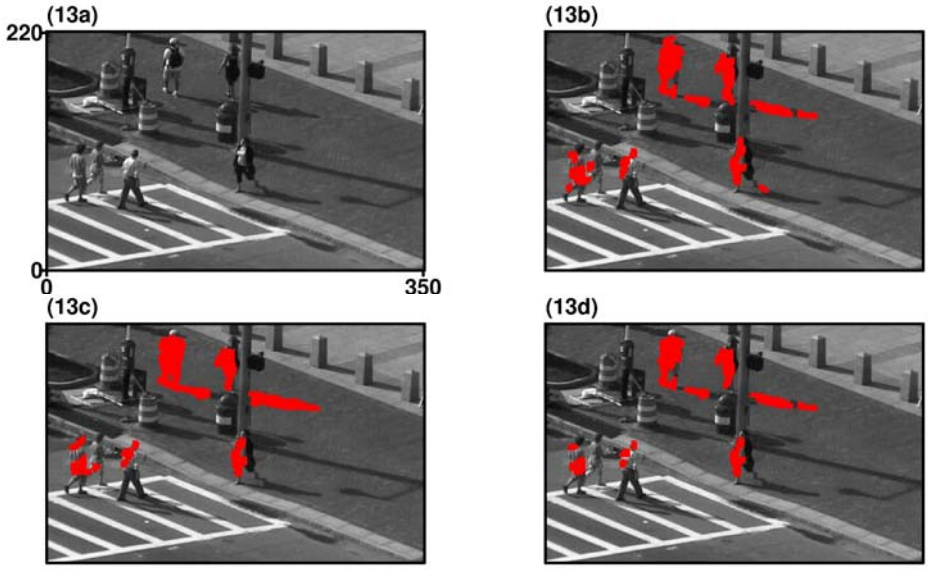

Figure 13 – Detection results for Sidewalk frame 292
(a) Camera frame 292. (b) Threshold exceedence map for normalized temporal differences. (c) Threshold exceedence map for normalized spatial differences. (d) Change detection map, based on minimum normalized differences. Detection fails for one of the six pedestrians present in this frame.

Over the 200-frame sequence (frames 101 – 300) that we analyzed for change, two frames had a combined total of three false alarms, and every pedestrian present was detected in 89% of the frames. Pedestrians were sometimes missed as they passed through pixels with high spatial variance. A typical case is shown in Figure 13, where one of the six pedestrians occupies pixels with steep intensity gradients due to the crosswalk, curb, and an oil stain at the edge of the road. Detection of this individual, whose clothing is about the same grey tone as the street, is intermittent until he steps across the curb and onto the darker and more uniform sidewalk.

7 – DISCUSSION

In this report, we have introduced a new method for real-time change detection in high-jitter environments. The method utilizes two separate statistical models for pixel intensity. The

first model relies on a temporal history of frame data, and attempts to capture the correlated structure of jitter-induced effects in a low dimensional background subspace. Temporal standard deviations are estimated from subspace projection residuals, and incorporate variability due to both camera jitter and natural scene elements. The second model employs spatial moments calculated from a single frame or background image. These estimates anticipate the effects of jitter on pixel intensities, and perform well for sudden increases in jitter magnitude not seen in the sequence of frames used for model initialization and training. The approach was originally designed for scenarios like the Bearcam sequences, in which the changes to be detected are small in both intensity and spatial extent. However, with suitable adjustment to the initialization process and a few of the tunable parameters, strong performance was also achieved for the challenging Sidewalk sequence.

Topics for continued research include extension of the basic technique to multiband data, application-specific parameter adjustment, and the development of improved methods for initialization. Adapting the current approach to RGB color or multispectral data has the potential to provide strong change detection capability across a broad range of sensors and applications, and is presently under investigation.

Selection of reasonable values for the various tunable estimation and detection parameters and thresholds requires some experience with the specific change detection problem at hand. For example, with stable background scenes decay parameter β_1 can be set to a larger value than would be appropriate for scenes (or sensors) that are subject to gradual drifts in illumination, pointing, or focus. If the changes to be detected are expected to be spatially extensive, increasing detection cluster size *NEIGH* can help to mitigate potential false alarms. Approximate knowledge of the jitter distribution assists the user in specifying a suitable value of the jitter standard deviation σ. If continuing detection of persistent change is desired, the target update rate β_2 can be increased to unity, while it can be reduced to zero if no target suppression is needed.

Rapid initialization of temporal and spatial moments in the presence of target energy is challenging, particularly when target changes occur close to scene gradients. Excellent results were obtained for the Sidewalk sequence with a two-step approach, in which initial moment

estimates computed without target suppression were used to suppress later targets when calculating a second set of estimates. An alternative strategy might be to reduce the target suppression threshold steadily throughout the training period, akin to an annealing schedule, gradually removing target change energy from the moment estimates.

The change detection algorithm presented in this paper has demonstrated real-time performance, along with robustness to naturally variable pixels, gradual scene change, and varying levels of camera jitter. With appropriate initialization and tuning of the estimation and detection parameters, it can be applied across a wide range of image types and problem domains.

ACKNOWLEDGMENTS

This research was supported by Laboratory Directed Research and Development, Sandia National Laboratories, U.S. Department of Energy, under contract DE-AC04-94AL85000. The zebra image of Figures 1, 2, and 5 was made available by the U.S. Fish and Wildlife Service, photographer Gary M. Stolz. The authors gratefully acknowledge the Woodland Park Zoo (Seattle, WA) for permission to publish the Bearcam frame sequences shown in Figures 7 and 9, and Professor Pierre-Marc Jodoin of the University of Sherbrooke for providing us with the Sidewalk frame sequence of Figures 11, 12, and 13. We also thank John Richards and John Feddema for their thoughtful reviews of the manuscript.

REFERENCES

[1] R. Badeau, B. David, and G. Richard, "Fast Approximated Power Iteration Subspace Tracking," *IEEE Trans. Signal Proc*., vol. 53, no. 8, pp. 2931-2941, 2005.

[2] P.E. Barry and M. Klop, "Jitter Suppression: A Data Processing Approach," *Proc. SPIE*, vol. 366, pp. 2-9, 1983.

[3] L. Bruzzone and R. Cossu, "An Adaptive Approach to Reducing Registration Noise Effects in Unsupervised Change Detection," *IEEE Trans. Geosci. Remote Sensing*, vol. 41, no. 11, pp. 2455-2465, 2003.

[4] R. Collins, A. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 745-746, 2000.

[5] P. Comon and G.H. Golub, "Tracking a Few Extreme Singular Values and Vectors in Signal Processing," *Proc. IEEE*, vol. 78, no. 8, pp. 1327-1343, 1990.

[6] X.G. Doukopoulos and G.V. Moustakides, "Fast and Stable Subspace Tracking," *IEEE Trans. Signal Proc.*, vol. 56, no. 4, pp. 1452-1465, 2008.

[7] M. Diani, A. Baldacci, and G. Corsini, "Novel Background Removal Algorithm for Navy Infrared Search and Track Systems," *Optical Engineering*, vol. 40, no. 8, pp. 1729-1734, 2001.

[8] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151-1163, 2002.

[9] J.R. Gleason, "Understanding Elongation: The Scale Contaminated Normal Family," *JASA*, vol. 88, no. 421, pp. 327-337, 1993.

[10] J.D. Hulsmann and P.E. Barry, "An Eigenvector Procedure for Eliminating Line-of-Sight

Jitter Induced Noise from Staring Mosaic Sensors," *Nineteenth Asilomar Conference on Circuits, Systems and Computers*, pp. 1-5, 1985.

[11] P.-M. Jodoin, J. Konrad, V. Saligrama, and V. Veilleux-Gaboury, "Motion Detection with an Unstable Camera," *IEEE Intl. Conf. on Image Proc*., 2008.

[12] P.-M. Jodoin, M. Mignotte, and J. Konrad, "Statistical Background Subtraction Using Spatial Cues," *IEEE Trans. Circuits and Syst. for Video Tech*., vol. 17, no. 12, pp. 1758-1763, 2007.

[13] N.L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions, Vol. 1*, 2nd Ed. New York: John Wiley & Sons, 1994.

[14] J.A. Kirk and M. Donofrio, "Principal Component Background Suppression," *IEEE Aerospace Applications Conference*, vol. 3, pp. 105-119, 1996.

[15] P.K. Maziaka, "Jitter Induced Clutter in Staring Sensors Arising from Background Spatial Radiance Gradients," *Optical Engineering*, vol. 21, iss. 5, pp. 872-881, 1982.

[16] A. Mittal, and D. Huttenlocher, "Scene Modeling for Wide Area Surveillance and Image Synthesis," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 160-167, 2000.

[17] A. Mittal, A. Monnet, and N. Paragios, "Scene Modeling and Change Detection in Dynamic Scenes: A Subspace Approach," *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 63-79, 2009.

[18] E. Oja and J. Karhunen, "On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix," *J. Math. Anal. and Appl*., vol. 106, iss. 1, pp. 69-84, 1985.

[19] N.L. Owsley, "Adaptive Data Orthogonalization," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 109-112, 1978.

[20] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, “Image Change Detection Algorithms: A Systematic Survey,” *IEEE Trans. Image Processing*, vol. 14, no. 3, pp. 294-307, 2005.

[21] T.D. Sanger, “Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network,” *Neural Networks*, vol. 2, pp. 459-473, 1989.

[22] K.M. Simonson, S.M. Drescher, Jr., and F.R. Tanner, “A Statistics-Based Approach to Binary Image Registration with Uncertainty Analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 112-125, 2007.

[23] C. Stauffer, and W.E.L. Grimson, “Learning Patterns of Activity Using Real-Time Tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747-757, 2000.

[24] A. Tartakovsky and R. Blažek, “Effective Adaptive Spatial-Temporal Technique for Clutter Rejection in IRST,” *Proceedings of SPIE*, vol. 4048, pp. 85-95, 2000.

[25] J.R.G. Townshend, C.O. Justice, C. Gurney, and J. McManus, “The Impact of Misregistation on Change Detection,” *IEEE Trans. Geosci. Remote Sensing*, vol. 30, iss. 5, pp. 1054-1060, 1992.

[26] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and Practice of Background Maintenance,” Proc. *IEEE Int. Conf. on Computer Vision*, vol. 1, pp. 255-261, 1999.

[27] J.W. Tukey, *Exploratory Data Analysis*. Reading, MA: Addison-Wesley, 1977.

[28] R.E. Williamson and H.F. Trotter, *Multivariable Mathematics: Linear Algebra, Calculus, Differential Equations*, 2nd Ed., Englewood Cliffs, NJ: Prentice-Hall, 1979.

[29] Woodland Park Zoo, “Bear Camera,” Jan. 2009; http://www.zoo.org/bearcam/cam.html.

[30] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, “Pfinder: Real-Time Tracking of the Human Body,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, p. 780-785, 1997.

[31] K. Wu, E. Otoo, and K. Suzuki, “Optimizing Two-Pass Connected-Component Labeling Algorithms,” *Pattern Anal. Applic.*, vol. 12, iss. 2, pp. 117-135, 2009.

DISTRIBUTION

1	MS 1208	Katherine Simonson, 5535
1	1208	Tian Ma, 5535
1	0576	John Feddema, 0576
1	0899	Technical Library, 9536 (electronic copy)
1	0123	D. Chavez, LDRD Office, 1011
1	0161	Legal Intellectual Property, 11500