Detecting Small, Moving Objects in Image Sequences Using Sequential Hypothesis Testing

Steven D. Blostein, *Member, IEEE*, and Thomas S. Huang, *Fellow, IEEE*

Abstract—**A new algorithm is proposed for the solution of an important class of multidimensional detection problems: the detection of small, barely discernible, moving objects of unknown position and velocity in a sequence of digital images. A large number of candidate trajectories, organized into a tree structure, are hypothesized at each pixel in the sequence and tested sequentially for a shift in mean intensity. The practicality of the algorithm is facilitated by the use of multistage hypothesis testing (MHT) for simultaneous inference, as well as the existence of exact expressions for MHT test performance in Gaussian white noise (GWN). These expressions predict the algorithm's computation and memory requirements, where it is shown theoretically that several orders of magnitude of processing are saved over a brute-force approach based on fixed sample-size tests. The algorithm is applied to real data by using a robust preprocessing procedure to eliminate background structure and transform the image sequence into a residual representation, modeled as GWN. Results are verified experimentally on a variety of video image sequences.**

I. Introduction

THE increasing importance of image sequence analysis raises fundamental research issues in multidimensional signal detection. We address the detection of small, barely discernible, moving objects in a sequence of digital images. Traditionally, frame-by-frame differencing and thresholding operations have been used for this purpose. However, their effectiveness diminishes in inverse proportion to object size and amount of object motion in the image plane. To this date, there has not been an effective decision-theoretic approach to the detection of small, low-contrast objects where the positions and velocities of the objects are not known. In the past, the problem has been simplified by projecting 3-D (space–time) data onto 2-D images. Unfortunately, projection results in significant performance loss in noisy backgrounds or where object motion is significant. Alternatively, an exhaustive search for moving objects in space–time requires a computationally infeasible matched filtering of thousands of candidate trajectories per pixel per image.

Manuscript received October 4, 1988; revised July 11, 1990. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant OGPIN-011 and the Army Research Office SDIO/IST Contract DAAL03-86-K-0111.

S. D. Blostein is with the Department of Electrical Engineering, Queen's University, Kingston, Ont. K7L 3N6, Canada.

T. S. Huang is with the Coordinated Science Laboratory, College of Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801-3082.

IEEE Log Number 9144714.

A. Applications

In astronomy, the image sequences consist mainly of sensor noise, as in the problem of using a mosaic charge coupled device (CCD) sensor at the output of a telescope for optically detecting meteors, satellites, or other small moving objects against a night-sky background. A similar problem is known as night-sky satellite surveillance, where the objective is to keep inventory of the increasing number of objects orbiting the earth [1]–[4]. Currently, long CCD exposure times are used to detect faint distant stars by suppressing the zero-mean noise present through temporal integration. This technique, effective for stationary objects, will yield suboptimally weak image plane streaks if the objects are moving.

In other applications, the images have highly structured backgrounds, as in the detection of dim moving targets on the earth's surface from a space-borne mosaic infrared sensor staring down at a fixed point on the ground [5]–[7]. In motion analysis of outdoor optical or forward-looking infrared scenes [8]–[10], the image sequence may contain drifting background clutter as caused by relative motion between the sensor array and terrain, ocean, or clouds.

In multiobject tracking, the computational burden of real-time processing from high resolution sensors poses challenging problems in radar processing [11]–[14]. In the past, rather than address the task of distinguishing objects from clutter directly, the problem was guised as a data association problem where a mixture of detected object points and clutter points are acquired by simplistic intensity centroid finding. This unnecessary multiple object tracking may be avoided by using more sophisticated object acquisition as a front-end to the tracking system.

B. Previous Work

An early approach to object detection, which is still employed in video coding applications to segment moving objects is to perform frame differencing followed by thresholding. In the case where the objects are small or where the noise level is high, a systematic procedure to decide whether or not an object is present is needed. The early techniques fall short of this goal: Cowart *et al.* [15]

use a Hough transform for detecting streaks in images; Rauch and Firschein [1] perform track assembly on binary images by successively combining the images in the sequence and then prescreening using 3×3 window operations. Here, the target must occupy exactly one pixel and the object must moves at a rate of exactly one pixel per frame. While computationally attractive, the ability to detect very dim targets is degraded compared to an approach that integrates intensity information over time. The detection of distinctive features in an isolated 2-D image has been addressed by Therrien *et al.* [16]. Watson and Woods [17] and Woods and Ekstrom [18] considered the detection of ocean eddies in satellite altimetry data.

It is well known that simple, easily analyzable detector structures result if the background noise is modeled as uncorrelated and/or Gaussian [19]. In reality, most scenes of interest have a more complicated statistical characterization. This has been addressed by prewhitening the background before detection, and has been proposed in [16] and [18]. However, in extending prewhitening to image sequences, the linear prediction and state estimation models used increase dramatically in complexity. Also, the assumed spatial statistical stationarity poses a problem in highly structured backgrounds.

In the past, object tracking in high noise situations was treated as a data association problem in a dense multitarget environment [11]. Multiple targets may not physically be present, but instead are due to false returns (clutter) from a sensor in the target's neighborhood, or validation region. Thus, the problem of reliable object acquisition, i.e., detection, is implicit to successful object tracking. As posed by Reid [12], Nagarajan *et al.* [14], Maybeck and Rogers [13] and in most other tracking references, the problem is one of state estimation in a dynamical system, augmented by additional procedures that account for data association and other uncertainty not considered by the dynamical model. When the number of target bins is large, as in the case of high resolution imagery, these tracking methods become computationally prohibitive. Indeed, combined detection and tracking seems to be a new trend in object tracking research. For example, Barniv [5] searches dim moving target paths using dynamic programming. Unfortunately, high computational requirements limit its use to very small image arrays.

In the case of image sequences, Chu [3] uses a 2-D, time-axis projection of the image sequence for data reduction. Object tracks are found by detecting faint line segments, or streaks, in the projection data. The drawback of time projection is evident in the case of an image sequence with object speeds exceeding one pixel per frame: assuming a constant-intensity object in additive noise, it is clear that the time-axis projection image has lower signal-to-noise ratio than any single image in the original sequence. A solution is to generalize to a set of projections which vary in space–time orientation. If enough projections are obtained, it is likely that one of these is well matched to the object's trajectory. Bruton and Bartley [20] use (3-D) resonant plane directional filtering in the spatial domain while Porat and Friedlander [7] employ a parallel bank of 3-D directional filters implemented in the frequency domain using a 3-D FFT. The output of each filter is a selectively enhanced 2-D image corresponding to a directional slice in space–time. Streaks corresponding to target tracks are then detected in each output image.

Mohanty [2] incorporates decision theory into the problem by finding a trajectory that maximizes a likelihood ratio formed by the object present/absent hypotheses directly on the 3-D data. A fixed number of frames are considered and processed in a batch manner. The background is assumed to have quasistationary Gaussian statistics that can be adaptively estimated. A fixed sample-size test is used. However, the method is too computationally expensive to be practical for real-time implementation.

In summary, the projection techniques sacrifice algorithm performance significantly, while the optimal matched filtering techniques have an impractically large search space. Therefore, the critical computational problem is to be able to perform some form of optimal matched filtering several orders of magnitude more efficiently. A technique to perform the matched filtering is a main contribution of this paper.

C. Overview

A straightforward generalization of statistical hypothesis testing to composite techniques with object location and velocity as unknown parameters results in prohibitive computation. In addition, optimal detector design is complicated by the nonstationary statistics of the given image sequence. A two-step paradigm is proposed: first, the redundant spatial and temporal information in the image sequence is removed. This prewhitening operation assumes that most of the image plane consists of background rather than object pixels. The object detection algorithm, optimized for a Gaussian white noise background model, is then applied to the prewhitened sequence.

To maximize performance, the detection algorithm should ideally find the set of pixels in space and time which contain the object's energy. This is achieved by hypothesizing a dense set of straight, constant velocity, pixel-sized trajectory segments originating at each pixel in each image. Even so restricted, the search space is too large to test a suitably dense set of candidate trajectories in a brute-force manner. Therefore, multistage hypothesis testing (MHT), is used to prune a tree-structured list of candidate trajectory segments at each pixel: summed grey levels are sequentially compared to two thresholds until either the hypothesis that the trajectory contains an object is rejected or accepted. When the test statistic falls between the two thresholds the decision is deferred and the trajectory state is stored in a list. Since the pixels contained in the first few levels of a hypothesis tree share many trajectories, early rejection (pruning) by the sequential procedure permits the tree-structured search to be

highly efficient. The MHT is designed via a truncated sequential probability ratio test [21]. Exact, closed-form expressions for analyzing the MHT for the case of independent (nonidentically distributed) Gaussian observations are derived in Section II-B and used to predict overall detection performance and computational requirements.

The paper is organized as follows: Section II provides the theoretical background on which the MHT detection algorithm is based, including a design procedure and performance analysis for the Gaussian white-noise case. The new algorithm described in Section III, including an analysis of its steady-state computation and memory requirements. Section IV addresses image sequence prewhitening, and experiments are presented in Section V.

II. Multistage Hypothesis Testing

In the following, assume that the mean object intensity is greater than and additive to the background noise, i.e., we are considering one-sided statistical testing procedures. If it is not known whether the objects are brighter or darker than the background, the algorithm can be run twice, once on the original sequence and an extra time on the same sequence after all pixels have been intensity inverted. Throughout this discussion, it is assumed that the total number of pixels belonging to the object is much less than the total number of pixels in the sensor array.

If an object's position in each image frame is known, then a maximally overlapping set of object pixels can be found, and a matched filter may be designed to perform the detection. This amounts to binary hypothesis testing of the collection of object pixels against a suitable threshold τ, for the absence (H_0) or presence (H_1) of the object. These tests are usually designed to satisfy certain criteria involving minimizing probabilities of error, such as false alarm and missed detection:

$$P(\text{choosing } H_1 | H_0 \text{ true}) \equiv \alpha$$
$$P(\text{choosing } H_0 | H_1 \text{ true}) \equiv 1 - \beta. \tag{1}$$

To provide context for the new results, fixed sample-size testing will be briefly reviewed. Section II-B motivates multistage hypothesis testing (MHT) by discussing the shortcomings of Wald's classical results [22]. Next, a procedure to design a truncated Wald test (a special case of the MHT) due to Tantaratana and Thomas [21] is summarized in Section II-B1. In Section II-B2, the small-sample MHT performance is analyzed: closed-form expressions are derived for the probability of the MHT reaching a given stage, as well as the MHT error probabilities as a continuous function of signal strength. These are the key results used to predict the performance and processing requirements of the MHT object detection algorithm presented in Section III.

A. Fixed Sample-Size Hypothesis Testing

Suppose we wish to test a collection of candidate object pixels for the presence of an object, represented statistically by a positive shift in the mean. The background noise distribution is zero-mean Gaussian and we want to test for a positive shift in the mean (background plus object): Let $\boldsymbol{x} \equiv x_1, x_2, \cdots$ be realizations of (i.i.d.) random variables $\boldsymbol{X} \equiv X_1, X_2, \cdots$. We denote λ as the common mean of each of the X_i's. Consider testing the hypothesis pair $\lambda = 0$ versus $\lambda = \lambda_1$, where λ_1 is the nominal mean used for test design purposes:

$$H_0: X_i \sim f(x - \lambda), \lambda = 0$$
$$H_1: X_i \sim f(x - \lambda), \lambda = \lambda_1 > 0 \tag{2}$$

for all i, where $f(x_i)$ is the zero-mean Gaussian probability density function of X_i, with variance ν^2. The Neyman–Pearson fixed sample size (FSS) test for (2) is obtained by testing K samples:

$$\sum_{i=1}^{K} z_i \begin{cases} \geq \tau \Rightarrow \text{choose } H_1 \\ < \tau \Rightarrow \text{choose } H_0 \end{cases} \tag{3}$$

where z_i is the observed realization of the random variable

$$Z_i = \ln\left[\frac{f(x_i - \lambda_1)}{f(x_i)}\right] = \lambda_1(X_i - \lambda_1/2)/\nu^2. \tag{4}$$

The first moment of the Z_i given λ is

$$\mu_\lambda \equiv E(Z_i | \lambda) = \lambda_1(\lambda - \lambda_1/2)/\nu^2 \tag{5}$$

the second moment is

$$m_\lambda \equiv E(Z_i^2 | \lambda) = \left(\frac{\lambda_1}{\nu}\right)^2 + \left(\frac{\lambda_1}{\nu}\right)^4 \left(\frac{\lambda}{\lambda_1} - \frac{1}{2}\right)^2 \tag{6}$$

and the conditional variance

$$\sigma_\lambda^2 = m_\lambda - \mu_\lambda^2 = \left(\frac{\lambda_1}{\nu}\right)^2 \equiv \sigma^2. \tag{7}$$

To reduce subscripts we define $\mu_1 \equiv \mu_{\lambda_1}$. Parameters K and τ are chosen to satisfy (2). For fixed sample-size tests, the error probabilities determine the threshold

$$\tau = K^{1/2}[\mu_1 \Phi^{-1}(\alpha) + \mu_0 \Phi^{-1}(1 - \beta)](\sigma/(\mu_0 - \mu_1)) \tag{8}$$

and number of test samples

$$K = [\Phi^{-1}(\alpha) + \Phi^{-1}(1 - \beta)]^2(\sigma/(\mu_1 - \mu_0))^2 \tag{9}$$

where Φ represents the standard (normalized) Gaussian distribution function. The inverse of Φ is denoted by Φ^{-1}.

In the object detection problem, the object position and velocity are unknown. In the absence of other *a priori* knowledge, there is little choice other than to resort to a suboptimal algorithm where binary hypothesis testing is used on a subset of the possible finite collections of image pixels. Each such collection corresponds to a segment of a hypothesized object trajectory. In this brute-force search, the number of necessary tests at each image pixel will typically be too large to consider performing exhaustive tests. Thus, we require an efficient technique by which to perform tests equivalent to (3).

B. Multistage Hypothesis Testing (MHT)

An alternative to the above FSS test is Wald's sequential probability ratio test (SPRT) [22] in which a test statistic formed from the x_i's is sequentially compared to an upper threshold, and a lower threshold. It turns out that any error probabilities can be achieved with an appropriate set of thresholds, and that the test will eventually terminate. As opposed to the FSS test of Section II-A, the sample size of the SPRT, is a random variable. It has been first shown by Wald and Wolfowitz [23] that the SPRT has the following optimality property: over a class of likelihood ratio tests, the SPRT minimizes $E(T|\lambda)$ at $\lambda = 0$ and $\lambda = \lambda_1$, where E denotes expectation, and T denotes the random stopping time of the sequential test. Therefore, employing an SPRT will reduce the average amount of per candidate trajectory computation.

An obvious disadvantage of the SPRT is that occasional long tests may result. The SPRT has an additional disadvantage [21] in that $E(T|\lambda)$ is sensitive to mismatch between actual object intensity λ and the nominal value λ_1. This problem is more significant if the test design error probabilities are small: under modest mismatch, the average test length may even exceed the size of an FSS test of equal power [24]. This mismatch between a hypothesized trajectory and the actual object's trajectory is an important limitation in the optimal matched filtering formulation of the object detection problem.

A practical compromise is a truncated SPRT [21], which maintains performance near the SPRT under optimal conditions, and is insensitive to situations involving parameter mismatch. This test differs from an SPRT in that a finite-stage truncation point exists. Truncated sequential probability ratio tests are an important example of two-threshold multistage tests: we employ the following.

Definition: A *multistage hypothesis test (MHT)* is any sequential test with a finite number of stages K.

In particular, for the Gaussian case discussed in Section II-A, we test the sum of observables

$$\sum_{j=1}^{i} x_j \begin{cases} \geq a_i & \Rightarrow \text{choose } H_1 \\ \geq b_i & \Rightarrow \text{choose } H_0 \\ \in (b_i, a_i) & \Rightarrow \text{go to stage } i+1 \end{cases}$$

in stage i, $1 \leq i < K$. At the Kth stage, we test

$$\sum_{j=1}^{K} x_j \begin{cases} \geq a_K \Rightarrow \text{choose } H_1 \\ else \quad \Rightarrow \text{choose } H_0 \end{cases} \tag{10}$$

where the a_i and b_i are thresholds at stage i.

1) MHT Test Design Issues: Given nominal $\hat{\alpha}$ and $1 - \hat{\beta}$ error probabilities, the goal is to design a suitable multistage test. From (10), we see that $2K - 1$ parameters must be specified for a K stage test. For even moderate K this number of free parameters is too large to design tests via nonlinear optimization. The number of design parameters can be reduced to four via a truncated SPRT [21] with boundaries $\hat{a}$, $\hat{b}$, and truncation point $\hat{\tau}$, at stage $\hat{K}$, at observation $i < \hat{K}$ we test

$$\sum_{j=1}^{i} z_j \begin{cases} \geq \hat{a} & \Rightarrow \text{choose } H_1 \\ \geq \hat{b} & \Rightarrow \text{choose } H_0 \\ \in (\hat{b}, \hat{a}) & \Rightarrow \text{take another sample.} \end{cases}$$

At $i = \hat{K}$ test

$$\sum_{j=1}^{\hat{K}} z_j \begin{cases} \geq \hat{\tau} \Rightarrow \text{choose } H_1 \\ < \hat{\tau} \Rightarrow \text{choose } H_0 \end{cases} \tag{11}$$

where the z_j's are defined in (4). The truncated SPRT can be viewed as a mixture of an SPRT and an FSS test [25]: if c_0 and c_1 are mixing constants on [0, 1], we set

$$\hat{a} = \ln\left[\frac{1 - (1 - c_1)(1 - \hat{\beta})}{(1 - c_0)\hat{\alpha}}\right] \tag{12}$$

$$\hat{b} = \ln\left[\frac{(1 - c_1)(1 - \hat{\beta})}{1 - (1 - c_0)\hat{\alpha}}\right] \tag{13}$$

$$\hat{K} = [\Phi^{-1}(c_0\hat{\alpha}) + \Phi^{-1}(c_1(1 - \hat{\beta}))]^2 \left(\frac{\sigma}{\mu_1 - \mu_0}\right)^2 \tag{14}$$

and

$$\hat{\tau} = \sqrt{\hat{K}}[\mu_1\Phi^{-1}(c_0\hat{\alpha}) + \mu_0\Phi^{-1}(c_1(1 - \hat{\beta}))]\left(\frac{\sigma}{\mu_0 - \mu_1}\right). \tag{15}$$

The design may be optimized by varying c_0 and c_1: values of c_0 and c_1 near 0 yield a test similar to the SPRT: low $E(T|\lambda)$ if $\lambda = 0$ or $\lambda = \lambda_1$ (no signal mismatch), and high $E(T|\lambda)$ otherwise. Alternatively, values of c_0 and c_1 near 1 result in similarity to an FSS test: higher $E(T|\lambda)$ but lower sensitivity to mismatch. In the test design, σ, μ_0, and μ_1 are known, and $\hat{\alpha}$ and $1 - \hat{\beta}$ are nominally chosen design values of the error probabilities. Once c_0 and c_1 are chosen, (11) is completely specified by $\hat{a}$, $\hat{b}$, $\hat{K}$, and $\hat{\tau}$. It has been shown [21], [25] that the resulting error probabilities are within the nominal design values: $1 - \beta \leq 1 - \hat{\beta}$ and $\alpha \leq \hat{\alpha}$

Remark 1: The truncated SPRT is not equivalent to performing an SPRT for the first $\hat{K} - 1$ stages, followed by performing an optimal FSS test at the $\hat{K}$th stage: in this case, α and β would be slightly greater than their design values [21].

Remark 2: For the Gaussian case considered here, we can express the truncated SPRT (11) as the following MHT: for $1 \leq i < K - 1$, we set $a_i = \hat{a}\nu^2/\lambda_1 + i\lambda_1/2$ and $b_i = \hat{b}\nu^2/\lambda_1 + i\lambda_1/2$. We also set $a_K = \hat{\tau}\nu^2/\lambda_1 + K\lambda_1/2$.

2) MHT Performance Analysis: In the following discussion, we derive exact expressions for $E(T|\lambda)$ and probability of choosing H_1 given λ, for the Gaussian case. For the SPRT, Wald [22] derives approximate expressions for the above quantities, where it is assumed that the SPRT exactly terminates on the threshold boundaries.

Wald's expression is thus a lower bound estimate of the actual error probabilities. These approximations are quite close to actual performance in cases where the error probabilities are reasonably high and where the number of observations are large. Recently, Tantaratana and Poor [25], [26] derived asymptotic expressions for the truncated SPRT for the limiting case of weak signals and large K. Unfortunately, these analyses are inaccurate for the object detection problem, where tests have very low error probabilities and a small number of stages.

In the following, the approach of Aroian and Robinson [27] is invoked, which is applicable since the number of stages is small. Since a minor error exists in [27], the corrected expressions are given below. We restrict the discussion to the case of interest where the noise is Gaussian; however, similar expressions can be derived for other probability distribution functions by the same technique. Let

$$f(x) = \frac{1}{\sqrt{2\pi\nu}} \exp\left[-\frac{1}{2}\left(\frac{x-\lambda}{\nu}\right)^2\right] \tag{16}$$

and $F(x) = \int_{-\infty}^{x} f(t)\,dt$. For $1 \le i \le K$, define

$$r_i(\lambda) \equiv \Pr(\text{MHT reaches } i\text{th stage} | X_i \sim f(x)), \text{ and} \tag{17}$$

$$\gamma_i(\lambda) \equiv \Pr(\text{accept } H_1 \text{ at } i\text{th stage} | X_i \sim f(x) \text{ and test reaches } i\text{th stage}). \tag{18}$$

First, the following quantities are initialized:

$$f_1 = f$$
$$r_1(\lambda) = 1$$
$$r_2(\lambda) = \int_{b_1}^{a_1} f(x)\,dx$$
$$\gamma_1(\lambda) = 1 - F(a_1).$$

Then, for $i = 1, 2, 3, \cdots$ and $b_{i+1} < w < a_{i+1}$, the recursive convolution is performed:

$$f_{i+1}(w) = \int_{b_i}^{a_i} f(w - x) f_i(x)\,dx. \tag{19}$$

After each iteration, the probability of the test continuing after stage $i + 1$, i.e., reaching stage $i + 2$, is given by the total area under f_{i+1}:

$$r_{i+2}(\lambda) = \int_{b_{i+1}}^{a_{i+1}} f_{i+1}(x)\,dx. \tag{20}$$

Given that the test reaches stage $i + 1$, the probability of accepting hypothesis H_1 can be calculated by evaluating

$$F_{i+1}(x) = \int_{-\infty}^{\infty} F(x - z)\,dF_i(z)$$

at $x = a_{i+1}$, subtracting from one, and normalizing, which yields

$$\gamma_{i+1}(\lambda) = 1 - \frac{\int_{b_i}^{a_i} F(a_{i+1} - x) f_i(x)\,dx}{r_{i+1}(\lambda)}. \tag{21}$$

The normalization in the rightmost expression differs from [27]. The above expressions are evaluated using standard numerical integration techniques. Fast convergence is achievable since the Gaussian density is a smooth function: usually 128 samples can provide six significant digits of accuracy. We remark that computationally attractive alternative expressions are found in [28], in which the above recursive convolution is approximated by the standard bivariate Gaussian cumulative distribution function.

Since the test must terminate at exactly one of the K stages, the power function consists of K mutually exclusive events. Thus the probabilities can be summed yielding

$$\Pr(\text{accept } H_1 \text{ at any stage} | X_i \sim f(x)) = \sum_{i=1}^{K} \gamma_i(\lambda) r_i(\lambda). \tag{22}$$

Thus, the actual false alarm rate, $\alpha = \sum_{i=1}^{K} \gamma_i(0) r_i(0)$, while the detection probability is given by $\beta = \sum_{i=1}^{K} \gamma_i(\lambda_1) r_i(\lambda_1)$. The average test length, $E(T|\lambda)$, is obtained by expressing the probability of termination in the ith stage as $r_i(\lambda) - r_{i+1}(\lambda)$, $1 \le i < K$, and termination in the Kth (final) stage with probability $r_K(\lambda)$. Thus,

$$E(T|\lambda) = \left[\sum_{i=1}^{K-1} i(r_i(\lambda) - r_{i+1}(\lambda))\right] + K r_k = \sum_{i=1}^{K} r_i(\lambda). \tag{23}$$

In Section III, the above performance analysis will be used to predict the memory and processing requirements of the MHT object detection algorithm.

III. A MHT Algorithm for Object Detection

We now apply the above one-dimensional results to an algorithm for detecting objects in (prewhitened) image sequences; a large number of simultaneous hypotheses represent candidate object trajectories. We show that multiple hypotheses can be organized in a hierarchical manner, permitting efficient evaluation using multistage hypothesis testing. First, the image detection problem is formulated in terms of searching lists of candidate trajectories. Next, the storage, representation, and steady-state processing of lists of undecided trajectories are discussed. A tree-structured organization of the hypotheses is then presented. The resulting algorithm is analyzed to derive performance, steady-state processing, and memory requirements on Gaussian white-noise backgrounds.

A. The MHT Object Detection Algorithm

To detect small objects moving in the image plane, we hypothesize candidate trajectory segments originating from each pixel in each image, and test each one using multistage hypothesis testing (MHT). Various restrictions are put on these candidate trajectory segments to control the search space, and affect both trajectory shape (straight line versus curved) and speed: we confine the search to short, straight trajectory segments (typically spanning less than 10 frames, or 1/3 of a second) that have constant object velocity. This is reasonable since many objects of interest are governed by simple kinematics. There is no inherent restriction to straight trajectories: if prior knowledge exists that objects follow curved trajectories, the search space may be modified accordingly.

We assume the object's image plane velocity (in pixels/frame), which depends upon the temporal image sampling, lies within a known upper bound. Otherwise, we consider the problem's search space to be too large for the algorithm to be applied. In addition, we assume that the image acquisition time is rapid enough so that the object is not smeared in individual images.

A very dense set of object trajectories is needed in order to minimize the performance loss due to mismatch between candidate trajectory pixels and actual object trajectory pixels. In order to test this dense set efficiently, MHT is combined with a tree-structured representation, as presented below.

1) Forming the Set of Candidate Trajectories: We first discuss the quantization of trajectory parameters to form the search space. In the following, a candidate trajectory refers to a several-pixel segment of an actual trajectory. More precisely, we use the following terminology:

Definition: A *real trajectory* of length K is a list of real-valued coordinates of the position of a hypothesized object's intensity center over a set of K imaging-time instants. The coordinate system is relative to the object's position at the first time instant.

Definition: A *discrete trajectory* segment of length K is a real trajectory whose coordinates have been quantized to integer pixel coordinates.

Note that an infinite number of real trajectories map to a finite set of discrete trajectories. In practice, given parameter ranges for object speed and direction and a required resolution, we generate a set of discrete trajectories in advance from a finite, closely spaced set of real trajectories. We call this the discrete test set and we denote its number of elements by D. A typical discrete test set used to process an image sequence is $K = 10$ pixels, speed $\in [0, 1]$ pixels/frame in 0.0002 pixels/frame increments, and a direction range of 360°, in 1° increments. In this case, the number of trajectories hypothesized at each pixel is $D = 4295$.

In each of the N^2 pixels occurring in each frame, the MHT algorithm initiates D, K-stage tests. The pixel intensities are assumed to fit a Gaussian model: each of the DN^2 tests sequentially compares the partial sums of the K pixel intensities along a hypothesized trajectory; the partial sums are compared to the upper and lower MHT thresholds until a decision is reached.

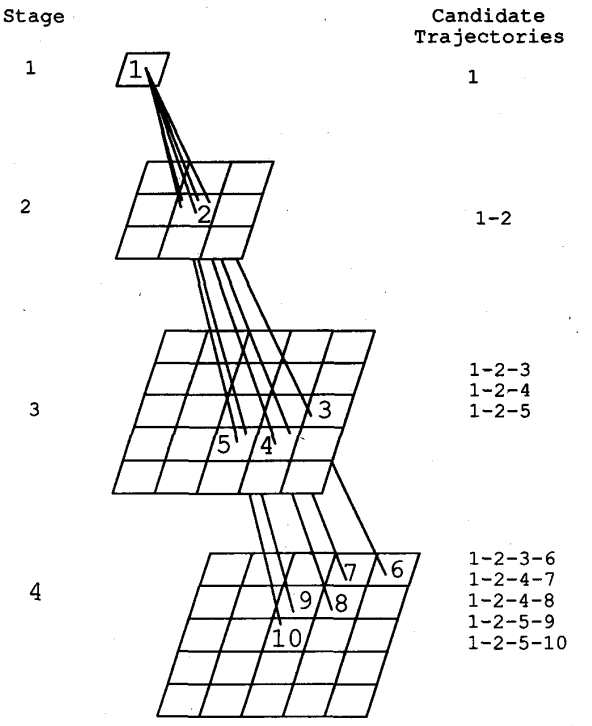


Fig. 1. A tree data structure is used to represent undecided trajectories. Each plane represents an image in a sequence. The lines represent five discrete trajectories. The trajectories would not extend beyond stage 2 if the sum of the grey levels at pixel locations 1 and 2 fall below the lower MHT threshold.

Due to the sequential nature of the MHT, tests may extend over a random number of images (up to a maximum of K) before a decision is reached. Thus, information of an undecided test's state must be temporarily stored from frame to frame. We store undecided tests (i.e., trajectories) in an indexed collection of lists, one for each of the N^2 pixels. Each undecided discrete trajectory is stored in the list corresponding to its starting location. This allows the storage of trajectories in relative coordinates.

2) Managing Undecided Tests: The discrete test set is highly redundant. For example, as shown in Fig. 1, the coordinates of the first trajectory pixel are shared by all trajectories in the test set, many trajectories have identical second-stage pixel coordinates, etc. It is wasteful to store each undecided trajectory independently. Instead, the intertrajectory pixel overlap can be exploited to derive a more efficient tree representation. Since many trajectories have common first parts, these trajectories can share a combined representation until they diverge. This results in significant savings because most trajectories are eliminated by MHT in the first few stages.

The tree representation of a discrete test set is implemented as a lookup table. The following information is stored with each node, i.e., lookup-table entry:

TABLE I
THE DISCRETE TEST SETS USED IN THE EXPERIMENTS

Name	5s0-1	10s0-1	10s0-2	17s0-0.3	20s0-1	10s0-20
# of stages	5	10	10	17	20	10
Minimum speed	0.0	0.0	0.0	0.0	0.0	0.0
Maximum speed	1.0	1.0	2.0	0.3	1.0	20.0
Speed resolution (pix./image)	0.004	0.002	0.01	0.0012	0.005	0.05
Minimum angle (rad)	0.0	0.0	0.0	0.0	0.0	−0.1
Maximum angle (rad)	6.28	6.28	6.28	6.28	6.28	0.1
Angular resolution (rad/image)	0.01	0.001	0.03	0.0012	0.0031	0.01
# of discrete trajectories	301	4295	406	1891	108	81
Total nodes in lookup table	461	11177	2581	9588	1716	682

1) The current test stage, i.e., the depth of the tree node.

2) x and y pixel offsets from the first-stage pixel.

3) Points to all nodes at the next stage that represent allowable continuations of discrete trajectories from the present node. This represents the divergence of trajectories that share common paths.

The lookup table is constructed off-line, given the following parameters for the test set: minimum speed, maximum speed, speed resolution, minimum angular direction, maximum angular direction, and angular resolution (see Table I). Details are shown in Fig. 2. Each undecided trajectory is identified by a tree node, which is simply an index into the lookup table, and a test statistic used by the MHT.

An example is given in Figs. 1 and 2, where the numbered pixel locations represent nodes. Entered alongside node 2 in the lookup table are stage number 2, pixel offsets (0, 0), and child nodes 3, 4, and 5. One of these children, node 5 contains stage number 3, pixel offsets (0, −1), as well as child nodes 9 and 10, both in stage 4, that are continuations of discrete trajectory segments passing through nodes 1, 2, and 5.

In summary, the tree-structured lookup table is constructed in advance from object direction and speed constraints. It allows the trajectory lists to store only node numbers (indexes into the look-up table, as shown in Fig. 2) and the test statistic values. Each node represents the root of an undecided trajectory subtree. The node contains the information necessary to propagate undecided trajectories to further stages. We show in Section III-C that tree search using MHT results in very efficient processing of candidate trajectories. The following is a list of steps that form the MHT object detection algorithm:

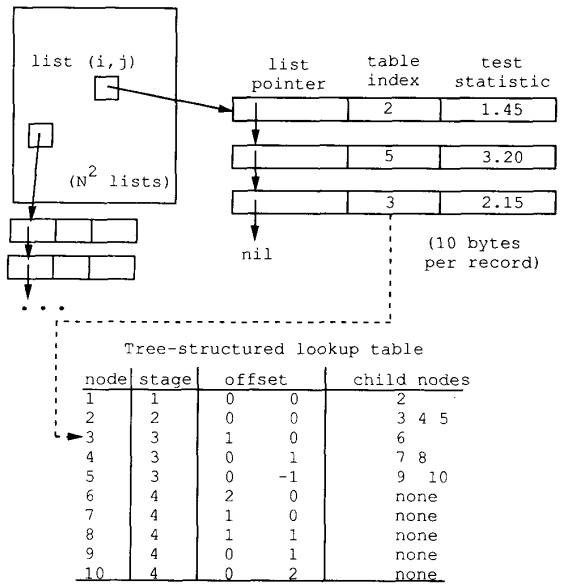


Fig. 2. Rather than store the trajectories individually, the N^2 lists represent undecided trajectory subtrees by indexing a tree-structured lookup table, which is constructed off-line.

MHT Object Detection Algorithm:

1) Construct a tree-structured lookup table off-line according to Section III-A2.

2) Initialize the N^2 trajectory lists as empty.

3) For each image in the (prewhitened) sequence:

For each pixel (i, j) in the image, $(i, j) \in [1, N] \times [1, N]$:

a) From trajectory list (i, j), get test statistic and lookup-table index.

b) Use table index to access current stage and trajectory extension information.

c) Extract each undecided trajectory from trajectory list (i, j) and extend it into the current image as determined by its node's children

d) For each extension, compute a new test statistic value by using (10) on the pixel grey levels.

e) Then apply an MHT at the current stage +1:

If below lower threshold, do nothing.

If above upper threshold, record the detected trajectory.

If (current stage +1) = K, do nothing.

Otherwise, add the extended, undecided trajectory, (next table index, new statistic) to list (i, j).

f) Apply the first stage of the MHT to pixel (i, j):
If below lower threshold, do nothing.
If above upper threshold, record the detected (single-pixel) trajectory.
Otherwise, add (index #1, grey level of pixel (i, j)) to list (i, j).

B. Algorithm Performance Analysis

1) Steady-State Memory and Computational Requirements: Using the MHT performance expressions derived in Section II-B2, the amount of per-pixel storage and computation can be calculated analytically for the case of a Gaussian white-noise background. This corresponds to the typical case of small objects where most candidate trajectories contain only background pixels. We proceed as follows: let $\rho(i)$ denote the number of stage i nodes in the discrete test set. Recall that $r_i(0)$ is the probability of the test reaching the ith stage given H_0. In the steady state (i.e., after processing at least K images) it may be easily verified that the expected number of tests performed per pixel is $\Sigma_{i=1}^{K} r_i(0)\rho(i)$ and that the expected number of nodes stored in the data base is $\Sigma_{i=1}^{K-1} r_{(i+1)}(0)\rho(i)$. This holds as long as the image data is statistically stationary. Image boundary effects have been neglected, which means that these expressions slightly overestimate the computation and memory requirements. Table II depicts $r_i(0)$ versus $\rho(i)$ for the example discrete test set of Section III-A2 and a 10-stage MHT (see Table III). Notice that $r_i(0)$ decreases with i faster than $\rho(i)$ increases. As a consequence of the constant velocity constraint, it can be shown that for large enough i, $\rho(i)$ becomes a constant while $r_i(0)$ decreases to zero exponentially. Using (20), Table IV shows the expected steady-state amount of processing needed for the MHT algorithm using the 10-stage test and the discrete test set described in Section III-A2. Results are compared against a brute force hypothesize-and-test algorithm where fixed sample-size tests are performed and each candidate trajectory is stored independently. As shown, several thousand distinct trajectories per pixel are searched with an average of only 35.04 additions and threshold tests per pixel. The expected list size in this case would be 12.53 undecided candidate trajectories per pixel.

2) Detection Performance: We now discuss the overall error performance of the MHT object detection algorithm: the overall false alarm rate and the overall detection performance. Under the same conditions as the previous calculation of the steady-state computation requirements, the steady-state number of false alarms can be shown to be $\Sigma_{i=1}^{K} r_i(0)\gamma_i(0)\rho(i)$. On the other hand, a brute-force algorithm based on a fixed sample-size (FSS) test of equivalent power to the above 10-stage multistage test, has $D\alpha$ false alarms in the steady state. It is interesting to note that the FSS-based algorithm will report more false alarms than the MHT-based algorithm. Intuitively, this is due to the ability of the latter to declare false alarms at early stages: from the MHT algorithm definition, a partial trajectory represents multiple distinct trajectories. Therefore, the MHT algorithm collapses the trajectories that contain false alarms into a smaller set.

TABLE II
NUMBER OF NODES VERSUS PROBABILITY OF REACHING A TEST STAGE

Stage	$\rho(i)$	$r_i(0)$
1	1	1
2	9	0.479
3	45	0.161
4	105	0.0550
5	301	0.0196
6	593	0.00722
7	987	0.00273
8	1752	0.001057
9	3089	0.000416
10	4295	0.000166

TABLE III
THREE SETS OF MULTISTAGE TEST PARAMETERS

# stages	5	10	17
Background variance, σ	1.0	1.0	1.0
Nominal object mean, λ_1	3.0	2.5	2.0
Nominal design value, $\hat{\alpha}$	1.0×10^{-6}	1.0×10^{-9}	4.0×10^{-10}
Actual false alarm prob., α	1.88×10^{-7}	3.84×10^{-10}	2.21×10^{-10}
Nominal design value, $\hat{\beta}$	0.80	0.90	0.95
Actual detection prob., β	0.93	0.95	0.97
Avg test length (under H_0)	1.29	1.73	2.71
Avg test length (under H_1)	3.68	7.31	11.7

The overall detection probability depends on the closeness of the match between the (unknown) set of pixels that contains the most object energy and the set of pixels belonging to one of the candidate trajectories. If the set of candidate trajectories is dense enough, then the effect of mismatch is insignificant. In fact, the need to reduce mismatch motivates a choice of dense discrete test sets. Assuming low mismatch, the overall detection probability of the algorithm would be equivalent to the detection probability of a single multistage test, β, which can be given by (22). We remark that the algorithm may typically have many space-time trajectories in which to detect the object. Thus, (22) serves as an approximate lower bound for the overall detection probability for the Gaussian case. We note that analogous to the reporting of false alarms, the number of detections reported will be smaller than in the case of an FSS-based algorithm of equivalent power.

Finally, the average detection time of the MHT algorithm is simply the average length of a multistage test given by (23), while for the FSS algorithm, the detection time is deterministic and identically equal to K.

IV. IMAGE SEQUENCE PREWHITENING

It is well known that both optimal and easily implementable detector structures are only obtainable for simple statistical models such as independent sequences of

TABLE IV
STEADY-STATE COMPUTATIONAL REQUIREMENTS AND PERFORMANCE

Algorithm	Memory (stored traj./pixel)	Computation (# tests/pixel)	False Alarms (per pixel)	Avg. Detection time (# images)
FSS				
10-pixel test	42950	42950	1.64×10^{-6}	10
expression	DK	DK	$D\alpha$	K
MHT				
10-stage test	12.53	35.04	1.31×10^{-6}	1.73
expression	$\Sigma_{i=1}^{K-1} r_{(i+1)}(0)\rho(i)$	$\Sigma_{i=1}^{K} r_i(0)\rho(i)$	$\Sigma_{i=1}^{K} r_i(0)\gamma_i(0)\rho(i)$	$\Sigma_{i=1}^{K} r_i(0)$

Gaussian random variables. In some applications, such as night-sky satellite surveillance, the image sequence is of a high noise environment, and Gaussian white noise is an acceptable model for the background. If the background is structured, statistical modeling becomes difficult and problem dependent. From a pragmatic point of view, a single detection algorithm is preferred for a variety of applications. This motivates the following paradigm:

1) Remove all background structure, i.e., transform the image sequence to an innovations sequence where all background pixels are i.i.d., standard Gaussian.

2) Apply an object detection algorithm, optimized for a Gaussian white noise background, to the innovations sequence.

A general treatment of image sequence prewhitening is beyond the scope of this paper. Instead, the particularly successful implementation used in the experiments will be discussed. Here, prewhitening is decomposed into temporal and spatial processing. Temporal prewhitening is accomplished by frame-by-frame differencing. This assumes that registration of the background is possible. In situations where there is too much background drift, temporal prewhitening is abandoned and only spatial techniques are used. In this case, it is likely that the effect of temporal correlation along object trajectories is insignificant, i.e., the background drift is often greater than the object's per frame image plane velocity. Therefore, pixels along candidate trajectories would be samples of distinct spatial areas of the drifting background, and can be considered to be uncorrelated.

Spatial prewhitening is achieved by nonstationary bias removal suggested by Cannon and Hunt [29], followed by space-varying variance estimation and normalization. The variance normalization procedure must have built-in robustness so that the effect of outlier influence (due to edges and other image structure) is minimized. In recent years, robustness in detection and estimation problems has been thoroughly studied [30]. It is well known that the sample variance is a poor estimate of scale when the underlying Gaussian distribution is contaminated with outliers. We therefore propose a median absolute deviation from the median (MAD) estimate of scale [31]. To make the MAD consistent with a zero mean, unit variance Gaussian distribution, we divide the MAD by $\Phi^{-1}(\frac{3}{4})$, where Φ^{-1} is the inverse of the standard normal distribution. We call this scaled estimate $\mathrm{MAD}_{N(0,1)}$. Associated robustness properties of the MAD estimate are discussed by Huber [31].

In applying robust technique to real images, the image is covered by a collection of spatially overlapping windows: the union of window pixels is the entire image. The collection is large enough so that each pixel lies in several distinct windows. For each window, the sample variance and $\mathrm{MAD}_{N(0,1)}$ estimates are computed. Then, we choose a "best" window for each pixel according to the following criteria: 1) the window must contain that pixel, and 2) among all such windows, we choose the window whose $\mathrm{MAD}_{N(0,1)}$ estimate is closest to its sample variance.

We remark that each pixel's window membership is determined by an empirical measure of a window's statistical similarity to a nominal Gaussian model. Windows that are split between regions (having nonunimodal histograms) tend not to be chosen. The resulting covering set of windows thus groups the pixels according to statistical similarity. The final step is to divide each pixel by its chosen patch's $\mathrm{MAD}_{N(0,1)}$ estimate.

After space-varying mean removal and scale normalization, the image intensities are limited to avoid spurious false alarms in the MHT object detection algorithm. As is common in robust techniques, we limit the intensities to lie within W standard deviations of the mean intensity.

An Example Implementation: In the experiments, the images are 128×128 pixels in size. If applicable, image registration and differencing (time decorrelation) are first performed. A 5×5, constant-coefficient averaging filter is then used to estimate and subtract out the space-varying mean. A set of square, 16×16 windows, placed every 8 pixels in both x and y image directions, are used to cover the image (4 patches/pixel). The MAD and sample variance are estimated from each window intensity histogram. Rather than using the sample median, the MAD's are estimated with respect to zero mean, as expected from space-varying mean removal (Section IV-B). Finally, a value of $W = 2.5$ standard deviations is used to trim the outliers. In Fig. 3(a), the original image of a road scene is shown. The result of temporal frame differencing and the removal of a space-varying mean is displayed in (b). In (c), the robust estimates of scale are shown: image brightness is proportional to the estimated intensity variance at a particular location. For example, the white area

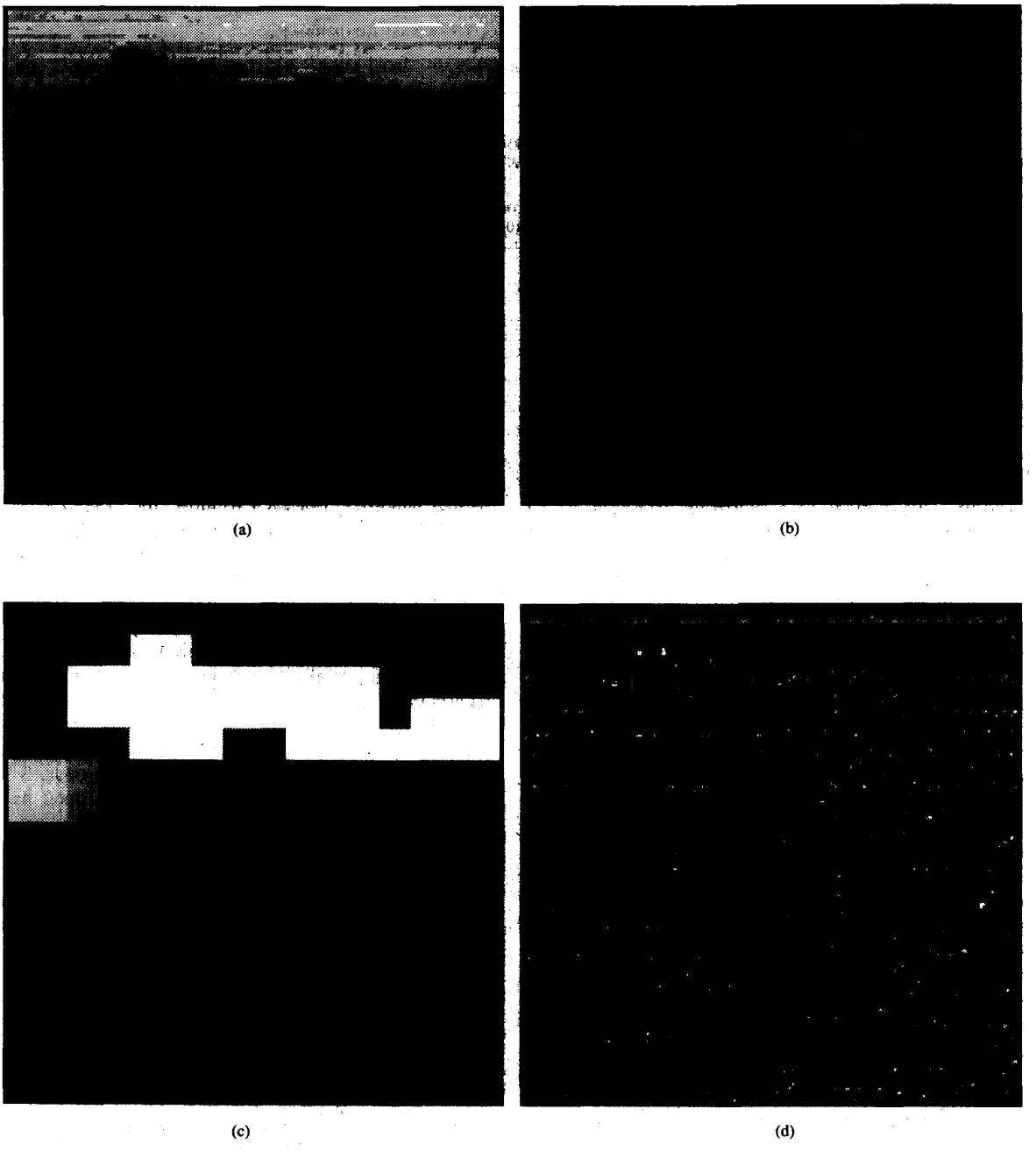


Fig. 3. (a) An image in a road scene sequence. (b) The result of frame differencing and space-varying mean removal. (c) Robust estimates of scale, proportional to image brightness. After variance normalization and outlier limiting, the prewhitened image, (d) results.

corresponds roughly to tree areas in (b), which consist of significant local intensity variation. Note that (c) performs a rough segmentation of the original image: the road, field, trees, and sky areas all have distinct scale estimates. After scale normalization and the suppression of outliers, the result of the prewhitening is shown in (d).

V. Experimental Results

A number of experiments have been performed in order to evaluate the MHT object detection algorithm and prewhitening procedure. The parameters of the discrete test sets used in the experiments are cataloged in Table I and will be referred to by the names given in the first row of

the table. The thresholds and error probabilities for the 5, 10, and 17 stage tests used in the experiments are given in Table III. The performance expressions presented earlier are used to determine the contents of this table. The first set of experiments, gives an indication of the algorithm's performance in ideal conditions: the background consists of synthetically generated Gaussian white noise. To measure absolute performance, a rough comparison to a human's ability to perform the same task is made in the detection of synthetic point targets in white noise, where the position and velocity of the targets are chosen randomly. A second experiment, tests the prewhitening steps discussed in Section IV, where the background consists of digitized photographs of drifting clouds, well spaced in time. The acquired data is temporally sparse enough so that image registration and subtraction are not possible. A third set of experiments tests the performance of the algorithm in outdoor scenes. Here, video images of distant, approaching vehicles on a roadway are used. The final set of experiments tests the algorithm in a night-sky, deep-space surveillance application using real data gathered by a telescope.

A. Performance in Synthetic White Noise: Ideal Conditions

A sequence of 80, 128 × 128 images of i.i.d. Gaussian random variables, of zero mean and unit variance, was generated. A set of 10 translating objects, each of intensity 2.7 and size 1.0-by-1.0 pixel, was added to this background. The objects were placed at real-valued locations within the image, determined by a uniform random number generator. Each object had a randomly chosen constant image plane velocity of magnitude no more than 1 pixel per frame. The object positions and velocities are given in Table V. The 80-frame sequence was quantized to 256 grey levels and displayed at 3.75, 7.5, 15, and 30 frames per second on a high-resolution monitor, repetitively, in a palindrome: (frame 1, 2, · · · , 80, 79, · · · , 2). Ten human subjects were allowed to look at the sequence at each of the four speeds for as long as they wished. None could detect any of the objects. (Some of the subjects were able to see one of the objects, #1, Table V, after being shown its trajectory on the monitor.)

First, the five-stage MHT, Table III, and the discrete test set, 5s0-1, Table I were used. Detailed analysis of the results revealed that all 10 objects were detected. However, at least 25 false alarms (detected trajectories that do not intersect actual object pixels) occurred, yielding an overall unsatisfactory performance. The same experiment was repeated using a more powerful 10-stage test, and the discrete test set, 10s0-1, Table I. The algorithm was able to detect 9 of the 10 objects successfully, usually detecting the objects repeatedly, giving an appearance of tracking. The results are displayed in Fig. 4: (a) shows image 20 in the sequence. (b) Shows the results of the algorithm after the first 20 images, in the form of a time-axis projection: the detected trajectories are shown in black, while the actual object trajectories are shown in gray. Figs. 4(c) and (d) show image 40 and the corresponding results up to image 40, (e) and (f) show the results up to the 60th frame, and (g) and (h) give the results up to image 80. In contrast to the previous experiment, there were no false alarms. The undetected object (Table V, #4) was inherently more difficult to detect since it moved out of the image boundaries at the 20th frame in the sequence. These results show conclusively that under conditions to which the MHT is optimized, humans are easily outperformed.

TABLE V
VELOCITIES OF SYNTHETIC OBJECTS USED

Object #	x Component (Pixels/Frame)	y Component (Pixels/Frame)	Initial x Position	Initial y Position
1	1.0	0.0	56.14	93.88
2	0.5	0.0	51.55	31.78
3	0.33	0.2	70.29	103.66
4	0.83	0.7	101.79	22.87
5	0.23	−0.3	82.09	38.36
6	−0.15	−0.6	34.63	65.03
7	−0.35	0.4	60.32	54.43
8	0.65	−0.1	40.74	68.29
9	0.95	0.19	52.92	48.63
10	−0.346	0.22	41.12	62.91

Finally, we remark that even in the absence of trajectory mismatch, some differences in detectability among pixel-sized objects occur due to subpixel location. Consider two pixel-sized objects, where the first object's intensity is divided among several pixels, while the second object's intensity is concentrated in a single pixel. Due to a lower ratio of object intensity to noise variance, it can be shown that the first object has a lower detection probability than the second object. For objects that contain many pixels, however, this effect becomes negligible.

B. Spatial Prewhitening: Drifting Cloud Backgrounds

A sequence of sixteen 128 × 128 images was extracted from a sequence of digitized photographs of a cloudy sky. The camera was stationary. The set of synthetic objects in Table V was added to the sequence. In this sequence, substantial temporal drift occurred from frame to frame, in addition to significant spatial grey-level variation. Only the objects in the darker regions of the images were detected by human subjects. The algorithm used the same MHT and discrete test set as the first experiment. However, all prewhitening steps of Section IV were used except frame differencing (due to the large change in background from frame to frame). No false alarms occurred and all objects were successfully detected. The results, and many others [28] show that the preprocessing was successful on data where image registration and subtraction is not possible.

C. Spatiotemporal Prewhitening: Natural Scenes

A sequence of 72 128 × 128 images was extracted from a 512-by-480 pixel road scene containing an approaching

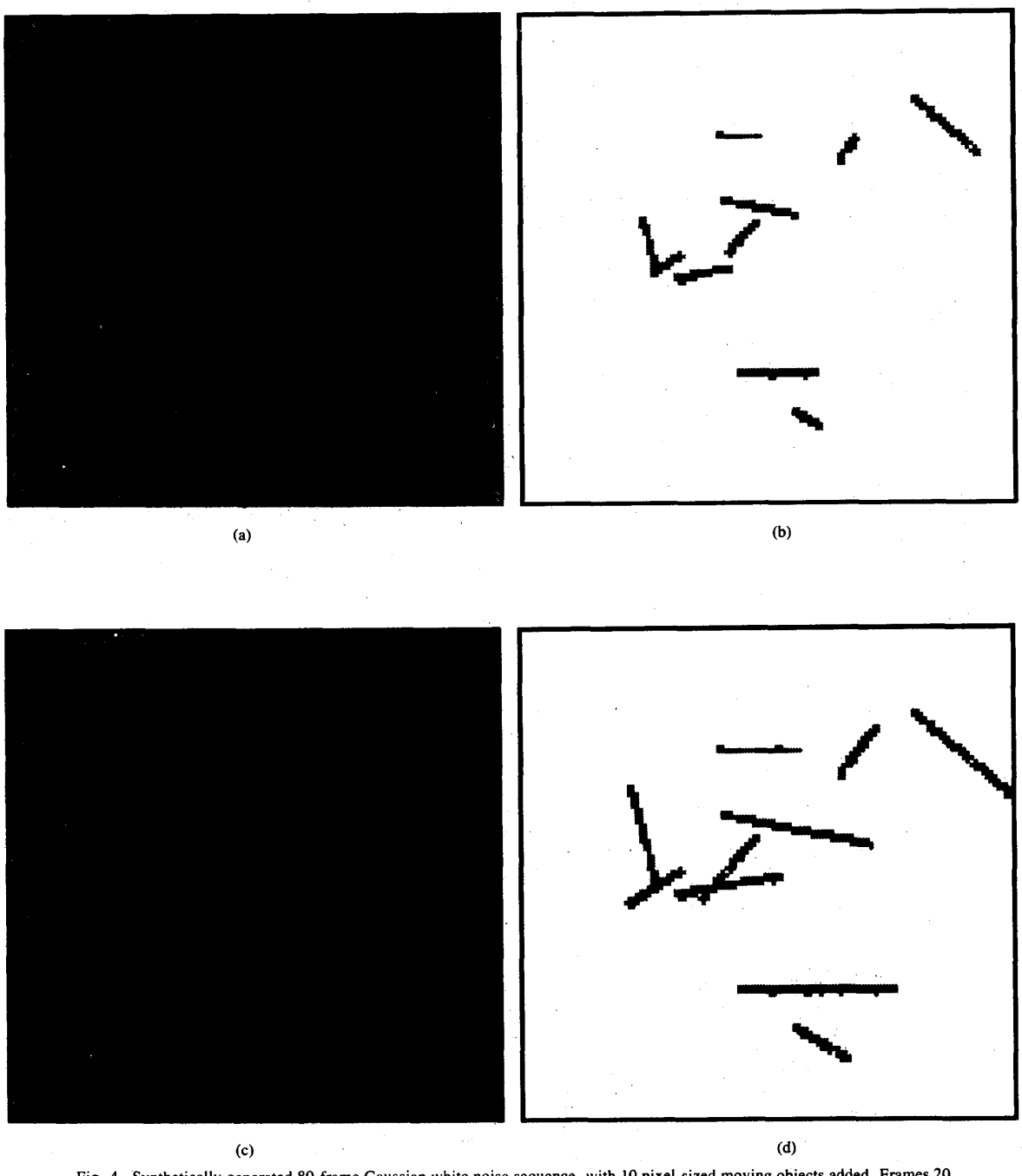
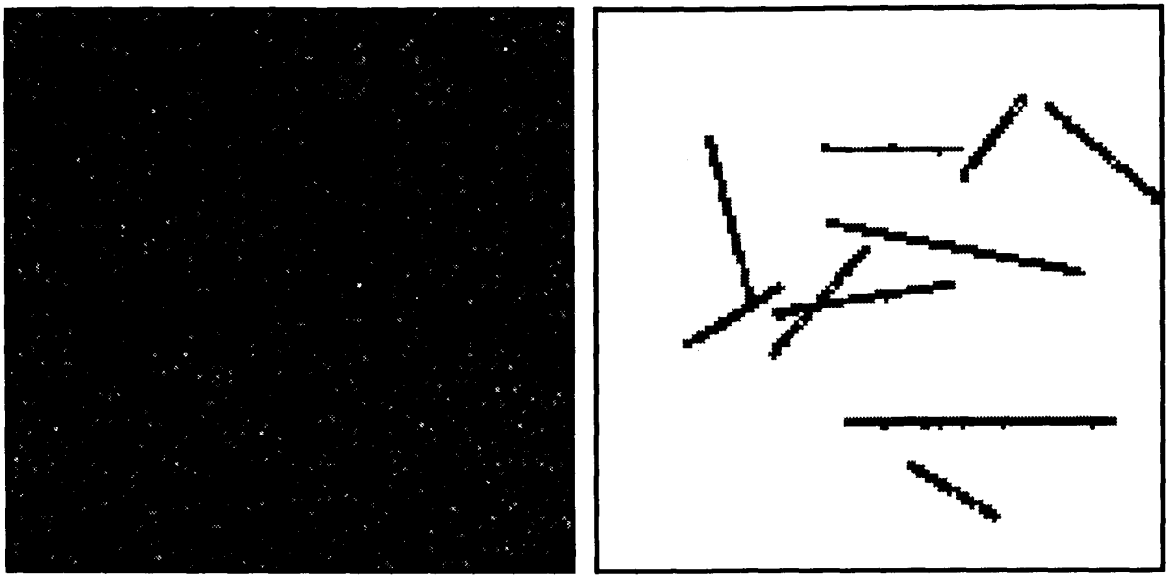
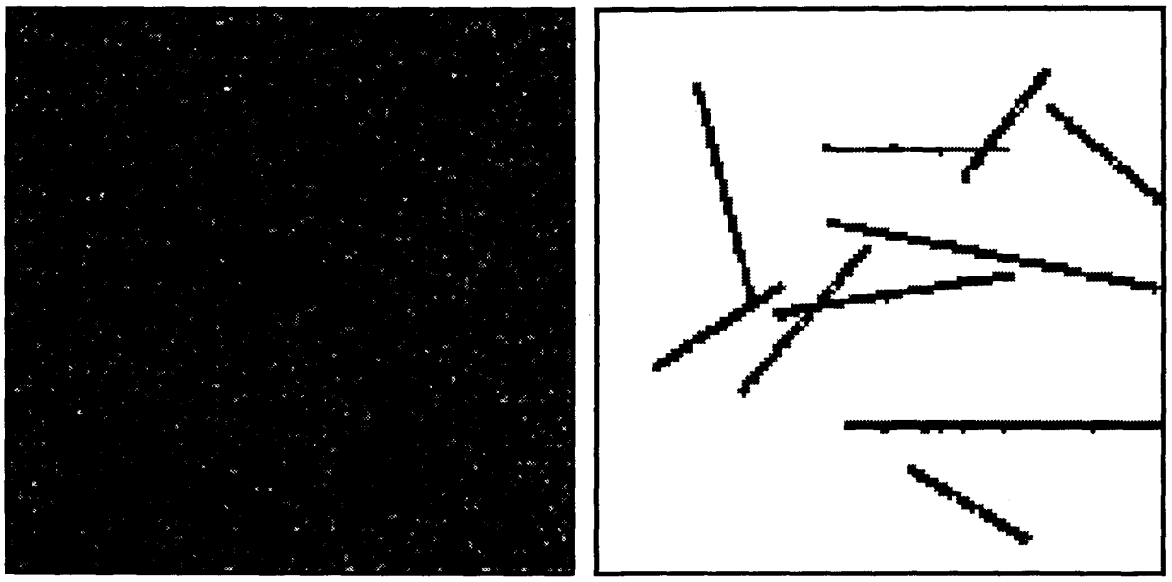


Fig. 4. Synthetically generated 80-frame Gaussian white noise sequence, with 10 pixel-sized moving objects added. Frames 20 and 40 are shown in (a) and (c), respectively. Actual object tracks are shown in gray, detected tracks in black. A 10-stage test is used with discrete test set 10s0-1, Table IV. (*Continued on next page.*)

runner. The sequence was digitized at about 6 frames per second, and one field per frame. Except for some trees swaying in the wind, the background and camera were both stationary, and frame differencing was used as an additional preprocessing step. Image 45 in the sequence was shown at the output of the prewhitening steps in Fig. 3 of Section IV. Tree movement caused strong edges, which confused the 10-stage MHT algorithm; six tree



(e) (f)



(g) (h)

Fig. 4. (*Continued.*) Frames 60 and 80 are shown in (e) and (g), respectively.

edges were detected in the sequence. These can be considered as false alarms in the sequence. To eliminate the detection of tree edges, the 10-stage test was replaced by a more powerful 17-stage test (Table III). The approaching runner is first detected in stage 8 of the 17-stage test, which occurs at image 45 in the sequence. With a longer (17 stage) decision time, the randomly fluctuating tree edges are discriminated from the steadily approaching object. A disadvantage, however, is the restriction of to constant trajectory velocities for 17 rather than 10 images.

In a similar background, a sequence of 140 128 × 128 images containing a car was digitized. The same prepro-

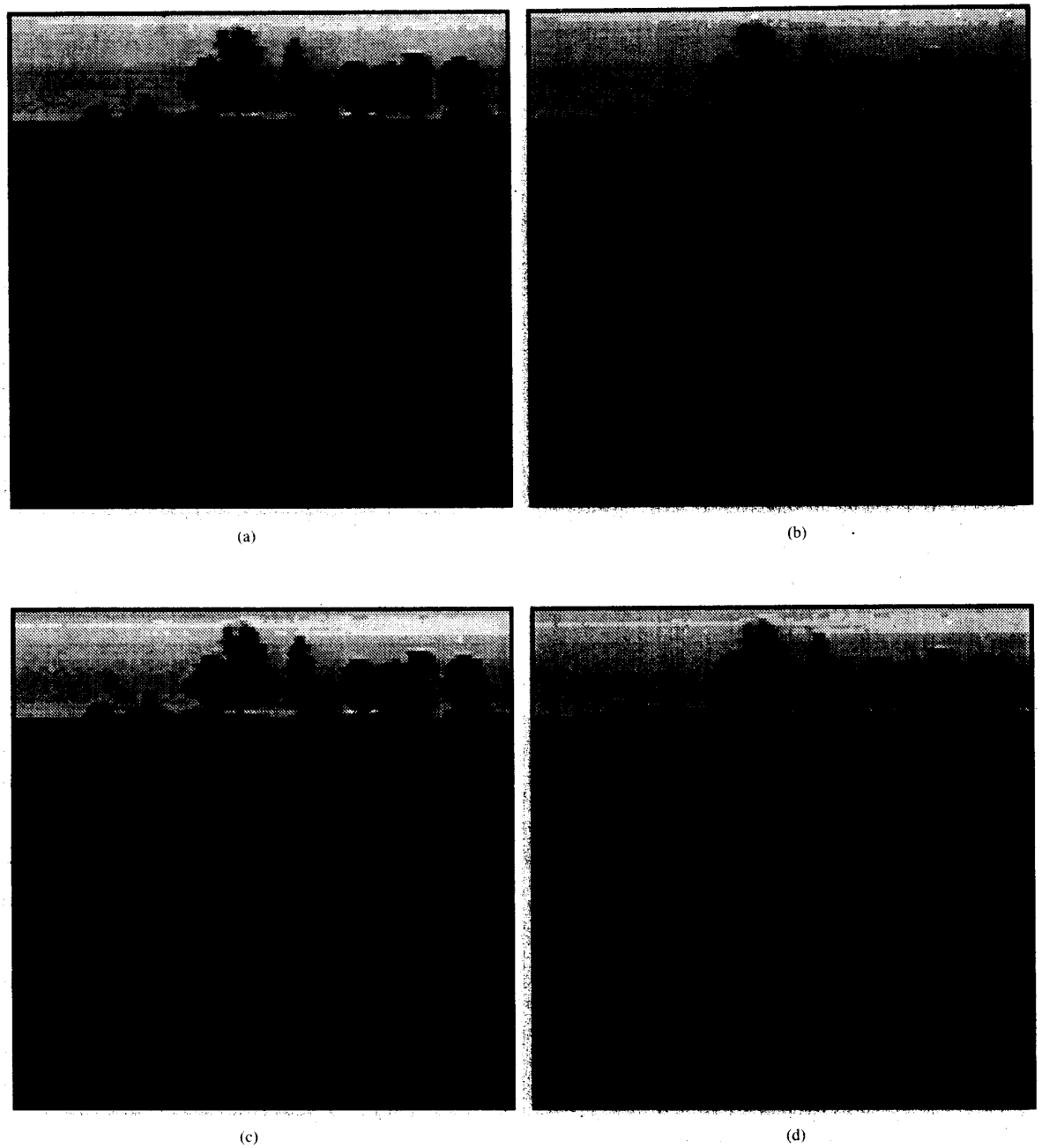
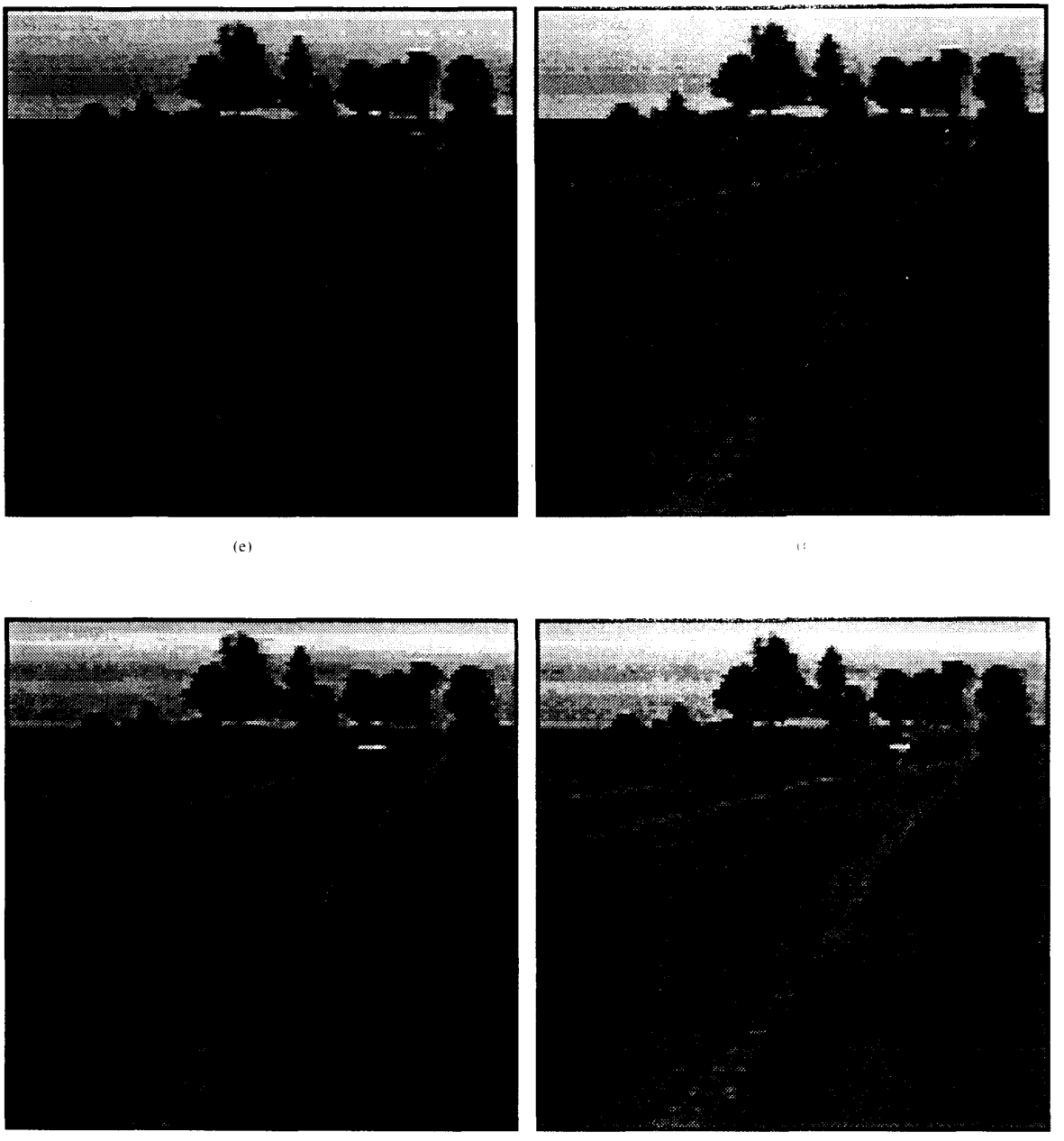


Fig. 5. A 140-frame road scene with an approaching car, digitized from a video camera. Frames 31 and 51 are shown in (a) and (c), respectively. Corresponding time projections of the results are shown in (b) and (d). Detected trajectories are shown in black, superimposed over the original images. A 17-stage test is used with discrete test set 17s0-0.3, Table IV. (*Continued on next page.*)

cessing and 17-stage image plane motion detection algorithm was used as in the previous case: the car was first detected in stage 16 of a 17-stage test in image 51. As before, no false alarms were reported. Figs. 5(a), (c), (e), and (g) show images 31, 51, 71, and 91 in the sequence, respectively. The detected trajectories up to each corresponding image are shown in black in (b), (d), (f), and (h) as time-axis projections.



(e) (f)

(g)

Fig. 5. (*Continued.*) Frames 71 and 91 are shown in (e) and (g), respectively. [illegible] are shown in (f) and (g).

D. Applications in Electrooptical Imaging of Deep Space

Image sequences of a 2.5° arc of the night sky were acquired at a rate of about 3 512 × 512 digital images per second.[1] It has been shown that the detection of small objects [illegible] other celestial bodies is [illegible] detection in a time-exposed [illegible] one if the objects are dim and [illegible] is high. To reduce the large volume of [illegible] (128 × 128) resolution images are formed by [illegible]. Of course, some performance [illegible] detection of very small ob-

[1]The data was provided courtesy of M. Axelrod, Lawrence Livermore Laboratory.

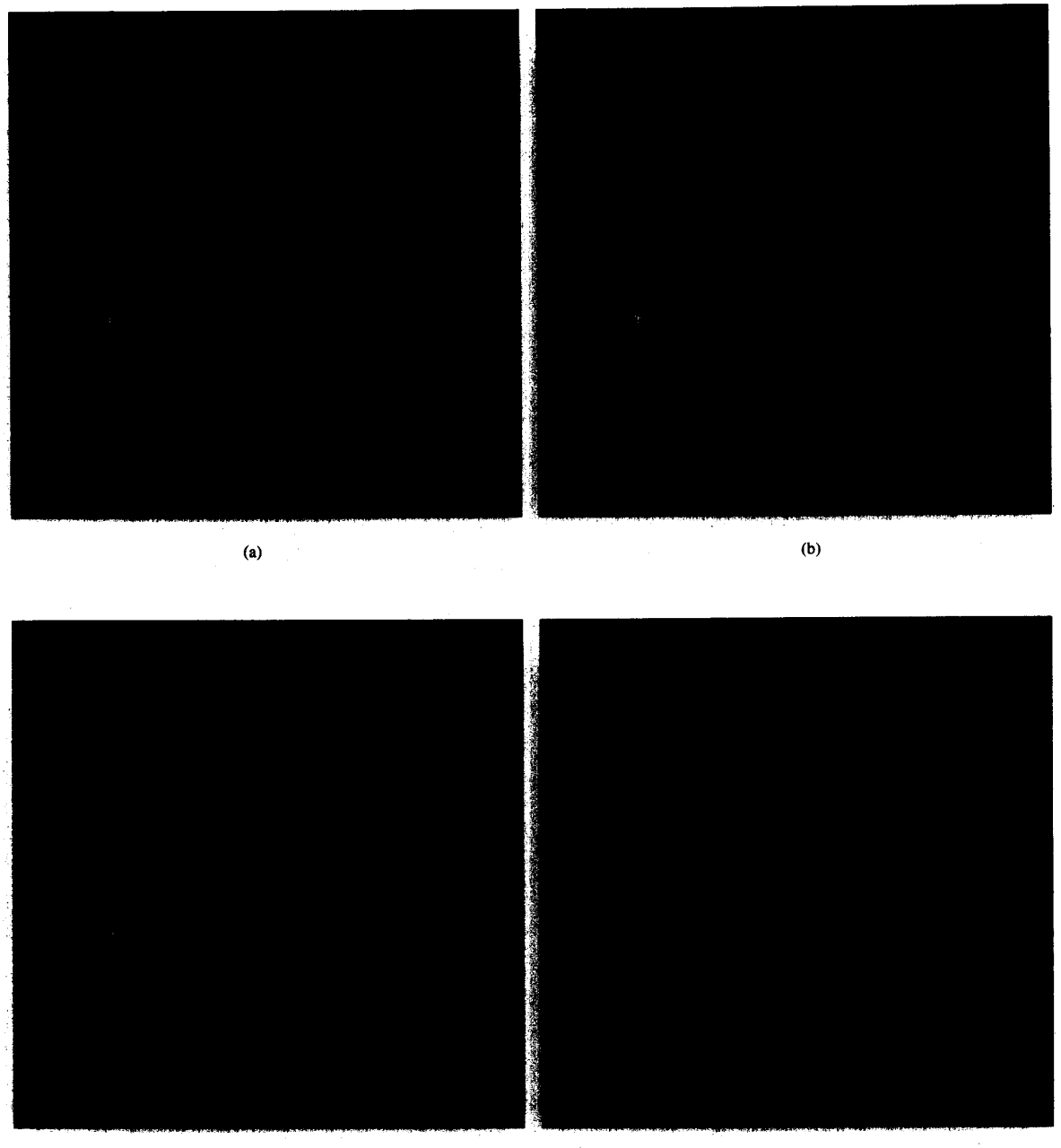
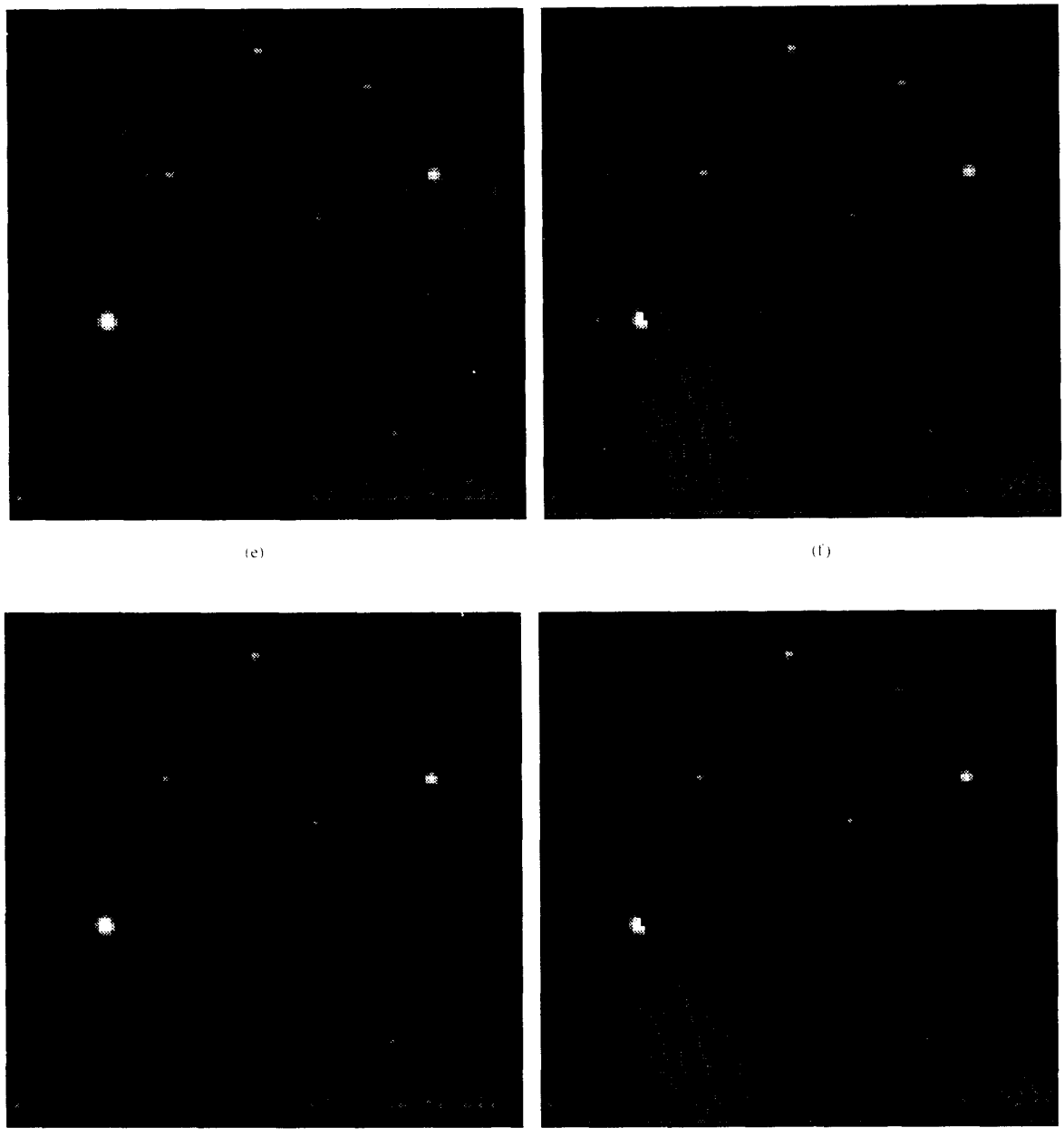


Fig. 6. A sequence of 78 images of the night sky, digitized by a CCD array at the output of a telescope. Frames 20 and 40 are shown in (a) and (c), respectively. Corresponding time projections of the results are shown in (b) and (d). Detected trajectories are shown in black, superimposed over the original images. A 17-stage test is used with discrete test set 10s0-2, Table IV. (*Continued on next page.*)

jects. By displaying the 78 image sequence as a palindrome at half the video rate, four objects moving at about 2 pixels per image in a vertical direction can be seen by humans. The sequence was processed using the standard prewhitening steps described earlier, including frame differencing. The MHT object detection algorithm was then applied using the 10-stage test and discrete test set, 10s0-2, Table I. The results are displayed in Figs. 6(a), (c), (e), (g) at four different time instances: images 20, 40, 60, and 78, respectively. The detected trajectories, in



(e) (f)

(g) (h)

Fig. 6. (*Continued.*) Frames 60 and 78 are shown in (e) and (g), respectively. Corresponding time projections of the results are shown in (f) and (h).

black, are superimposed on the original images in (b), (d), (f), and (h). As shown, the trajectories are repeatedly detected, giving the appearance of object tracking.

The experiment was repeated on a 128 × 128 subsection of a 512 × 512 sequence of 100 images. The same algorithm parameters were used as before. In this case, the object was dimmer and smaller than previously. As a result, only one segment of the object's trajectory was found by the algorithm. This sequence highlights the difference between object detection and object tracking. Once the object is detected, a separate tracking procedure may be invoked.

E. Computational Requirements

The steady-state number of undecided trajectories in the first night-sky image sequence (corresponding to Fig. 6)

averaged 240 000 for the 128 × 128 images in this example, while the predicted performance expressions from Section III-C1 yield a prediction of 258 000 steady-state trajectories. The small discrepancy can be explained by the small number of trajectories that continually exit the imaging area in the experiment, which is not taken into account in the analysis. In the runner sequence, 57 000 undecided trajectories were measured in the steady state. In this case, the predicted performance expressions predicted approximately 160 000 trajectories. The very large discrepancy is due to the prewhitened sequence having a statistical description that is not close to the Gaussian white noise model assumed in the analytical performance expressions.

VI. Discussion

In this paper, an algorithm for the detection of small, moving objects in image sequences has been presented. It is assumed that the background image pixels can be modeled as i.i.d. Gaussian random variables. The basic idea behind the tree-structured, MHT object detection algorithm is to efficiently search a dense set of space–time trajectory segments at every pixel in every image of the sequence. The algorithm performance, memory, and processing requirements are analyzed in Section III-C. The prohibitive amount of per-pixel computation is reduced several orders of magnitude compared to previous brute-force, trajectory-matching algorithms. This computational efficiency is accomplished by using multistage hypothesis testing, combined with a hierarchically organized trajectory list, to search the dense set of overlapping, multiple hypotheses.

The challenge in this statistical technique is for the algorithm to perform successfully in complicated and highly structured backgrounds. This has led to the two-step algorithm: image sequence prewhitening, followed by the MHT object detection algorithm, optimized for an i.i.d. Gaussian background model. The prewhitening consists of temporal as well as spatial techniques. Temporal prewhitening is accomplished by adjacent frame registration and subtraction, while spatial prewhitening is performed via robust space-varying mean and variance normalization. The combination of the two techniques has produced good results on a variety of low-quality, digitized video imagery.

The performance analysis of Section II is necessary to predict not only error rates (Section III-C2) but algorithm processing requirements by calculating the number of undecided trajectory segments per pixel that must be processed in the steady state assuming a Gaussian white noise background (Section III-C1). It turns out that the analysis is particularly accurate in cases where the images are noise-like at the output of the prewhitening step, such as in the case of image sequences of deep space. In more highly structured backgrounds, such as in the road scene, the amount of processing is significantly less than that predicted by the Gaussian model. A possible explanation is the presence of residual correlation in the image sequence. However, the Gaussian model is still a useful predictor of worst case performance in situations where the image sequences have maximum entropy, i.e., least spatiotemporal organization.

The research presented here may impact other areas in multidimensional signal detection. For example, a similar technique can be applied to feature detection problems in 3-D spatial data, as encountered in medical imaging or in nondestructive testing. Incorporating communication between trajectory lists may facilitate the detection of higher level image features such as moving edges, or boundaries between moving objects, which is applicable to low-bit-rate video coding and computer vision. Finally, the detection algorithm discussed can be used as a front end to multiobject tracking systems operating in cluttered environments.

References

[1] H. E. Rauch and O. Firschein, "Automatic track assembly for thresholded infrared images," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 253, pp. 75–85, 1980.

[2] N. C. Mohanty, "Computer tracking of moving point targets in space," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. PAMI-3, no. 5, pp. 606–611, 1981.

[3] P. L. Chu, "Optimal projection for multidimensional signal detection," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, no. 5, pp. 775–786, 1988.

[4] R. Holben, "An mti (moving target indicator) algorithm for passive sensors," in *SPIE Tech. Symp.*, 1980.

[5] Y. Barniv, "Dynamic programming solution for detecting dim moving targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-21, no. 1, pp. 144–156, 1985.

[6] A. Margalit, I. S. Reed, and R. M. Gagliardi, "Adaptive optical target detection using correlated images," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-21, no. 3, pp. 394–405, 1985.

[7] B. Porat and B. Friedlander, "A frequency domain algorithm to multiframe detection and estimation of dim targets," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 12, no. 4, pp. 398–401, 1990.

[8] J. S. Lee and C. Lin, "A novel approach to real-time motion detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn.*, 1988, pp. 730–735.

[9] N. Ayache and O. Faugeras, "Building, registering, and fusing noisy visual maps," in *IEEE Soc. Proc. First Int. Conf. Comput. Vision*, 1987, pp. 73–82.

[10] T. S. Huang, Ed., *Image Sequence Processing and Dynamic Scene Analysis*. Springer, 1983.

[11] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Trans. Automat. Contr.*, vol. 23, no. 4, pp. 618–626, 1978.

[12] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Contr.*, vol. AC-24, no. 6, pp. 843–854, 1979.

[13] P. S. Maybeck and S. K. Rogers, "Adaptive tracking of multiple hot-spot target ir images," *IEEE Trans. Automat. Contr.*, vol. AC-28, no. 10, pp. 937–943, 1983.

[14] V. Nagarajan, R. N. Sharma, and M. R. Chidambara, "An algorithm for tracking a maneuvering target in clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-20, no. 5, pp. 560–572, 1984.

[15] A. E. Cowart, W. E. Snyder, and W. H. Ruedger, "The detection of unresolved targets using the hough transform," *Comput. Vision, Graph., Image Processing*, vol. 21, pp. 222–238, 1983.

[16] C. W. Therrien, T. F. Quatieri, and D. E. Dudgeon, "Statistical model-based algorithms for image analysis," *Proc. IEEE*, vol. 74, no. 4, 1986.

[17] T. M. Watson and J. W. Woods, "Automatic detection of ocean ring signals," RPI Image Processing Lab., Tech. Rep. IPL-TR-81-018, 1981.

[18] J. W. Woods and M. P. Ekstrom, Eds., "Image detection and estimation," in *Digital Image Processig Techniques*, New York: Academic, 1984.

[19] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer, 1988.
[20] L. T. Bruton and N. R. Bartley, "The enhancement and tracking of moving objects in digital images using adaptive three-dimensional recursive filters," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 604–612, 1986.
[21] S. Tantaratana and J. Thomas, "Truncated sequential probability ratio test," *Inform. Sci.*, vol. 13, pp. 283–300, 1977.
[22] A. Wald, *Sequential Analysis*. New York: Wiley, 1947.
[23] A. Wald and J. Wolfowitz, "Optimum character of the sequential probability ratio test," *Ann. Math. Stat.*, vol. 19, no. 3, pp. 326–339, 1948.
[24] R. Bechhofer, "A note on the limiting relative efficiency of the wald sequential probability ratio test," *J. Amer. Stat. Ass.*, vol. 55, pp. 660–663, 1960.
[25] S. Tantaratana and H. V. Poor, "Asymptotic efficiencies of truncated sequential tests," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 6, pp. 911–923, 1982.
[26] S. Tantaratana and H. V. Poor, "Asymptotic relative efficiencies of multistage tests," *IEEE Trans. Inform. Theory*, vol. IT-31, no. 5, pp. 710–715, 1985.
[27] L. Aroian and D. Robison, "Direct methods for exact truncated sequential tests of the mean of a normal distribution," *Technometrics*, vol. 11, no. 4, pp. 661–675, 1969.
[28] S. D. Blostein, "A sequential hypothesis testing approach to detecting small, moving objects in image sequences," Ph.D. dissertation, Univ. Illinois at Urbana-Champaign, 1988.
[29] B. R. Hunt and T. M. Cannon, "Nonstationary assumptions for Gaussian models of images," *IEEE Trans. Syst., Man, Cybern.*, pp. 876–882, 1976.
[30] S. Kassam and H. V. Poor, "Robust techniques for signal processing: A survey," *Proc. IEEE*, vol. 73, no. 3, pp. 433–481, 1985.
[31] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

Steven D. Blostein (S'83–M'88) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1983, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1985 and 1988, respectively.

During 1981–1983, he was employed by Canadian Marconi Company and Bell-Northern Research. He has also performed consulting work for industry and government in the areas of document image compression, autonomous vehicle navigation, and image processing for tracking telescopes. Since 1988, he has been an Assistant Professor at Queen's University in the Department of Electrical Engineering. His current interests lie in sequential decision theory and image sequence analysis.

Thomas S. Huang (S'61–M'63–SM'76–F'79) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, China, and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge.

He was on the Faculty of the Department of Electrical Engineering at M.I.T. from 1963 to 1973; and on the Faculty of the School of Electrical Engineering, Purdue University, and Director of its Laboratory for Information and Signal Processing from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now Professor of Electrical and Computer Engineering and Research Professor at the Coordinated Science Laboratory. During his sabbatical leaves he worked at the M.I.T. Lincoln Laboratory, the IBM Thomas J. Watson Research Center, and the Rheinishes Landes Museum in Bonn, West Germany, and held Visiting Professor positions at the Swiss Institutes of Technology in Zürich and Lausanne, the University of Hannover in West Germany, and INRS-Telecommunications of the University of Quebec in Montreal, Canada. He has served as a consultant to numerous industrial firms and government agencies both in the U.S. and abroad. His professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has written 10 published books, and over 200 papers in network theory, digital filtering, image processing, and computer vision.

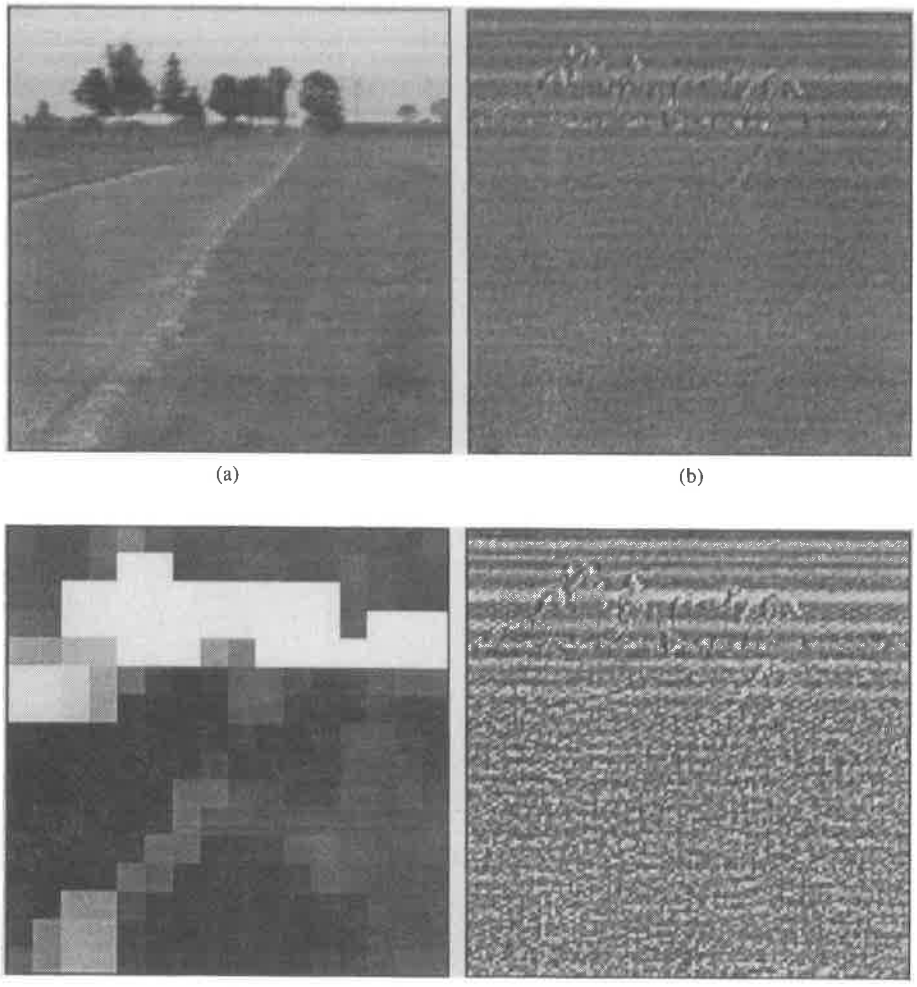
Dr. Huang is a Fellow of the Optical Society of America. He received a Guggenheim Fellowship (1971 to 1972), an A. V. Humboldt Foundation Senior U.S. Scientist Award (1976 to 1977), and a Fellowship from the Japan Association for the Promotion of Science (1986). He is an Editor of the *International Journal of Computer Vision, Graphics, and Image Processing*; and Editor of the Springer Series in Information Sciences, published by Springer Verlag; and Editor of the Research Annual Series on Advances in Computer Vision and Image Processing published by the JAI Press.

Corrections to "Detecting Small, Moving Objects in Image Sequences Using Sequential Hypothesis Testing"

Due to a production error, Figs. 3–6 on pages 1620–1627 of the above paper[1] did not clearly depict the original images. The figures have been reprinted to show the images more accurately.

Manuscript received July 29, 1991.
IEEE Log Number 9103940.

[1]S. D. Blostein and T. S. Huang, *IEEE Trans. Signal Processing*, vol. 39, no. 7, pp. 1611–1629, July 1991.



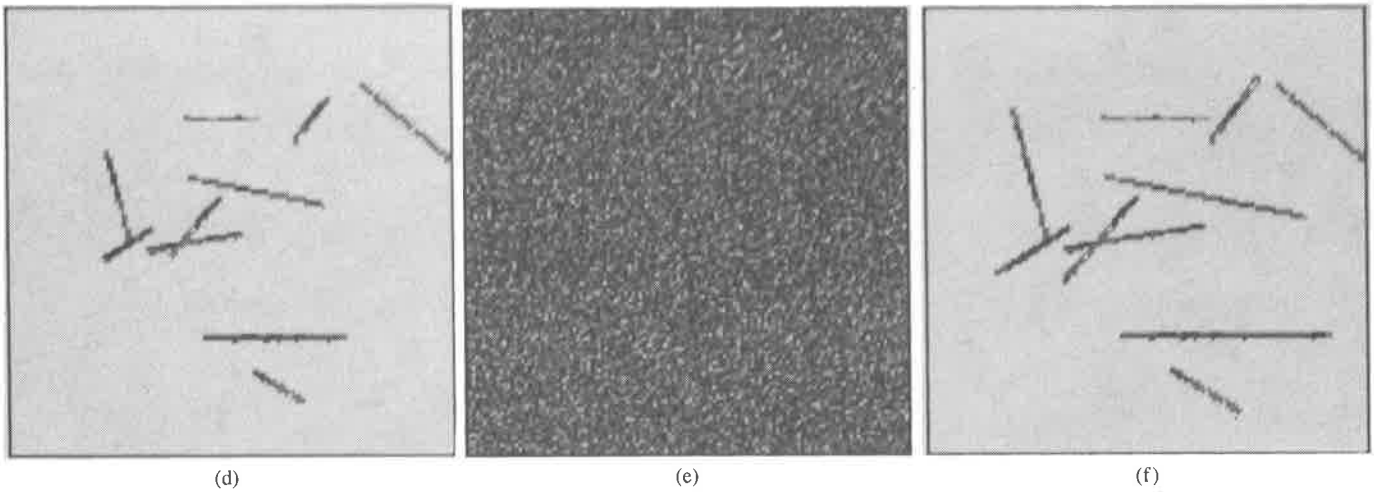
(a) (b)

(c) (d)

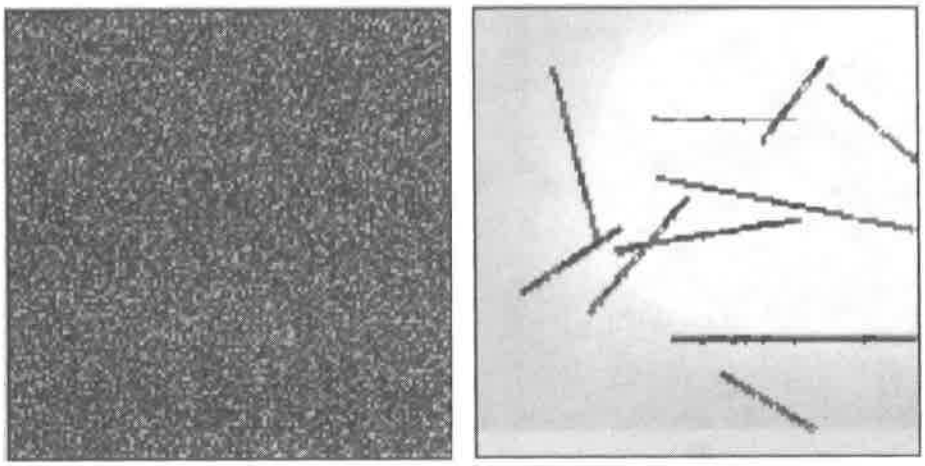
Fig. 3. (a) An image in a road scene sequence. (b) The result of frame differencing and space-varying mean removal. (c) Robust estimates of scale, proportional to image brightness. After variance normalization and outlier limiting, the prewhitened image, (d) results.

(a) (b) (c)

Fig. 4. Synthetically generated 80-frame Gaussian white noise sequence, with 10 pixel-sized moving objects added. Frames 20 and 40 are shown in (a) and (c), respectively. Actual object tracks are shown in gray, detected tracks in black. A 10-stage test is used with discrete test set 10s0-1, Table IV. *(Continued on next page.)*



(d) (e) (f)



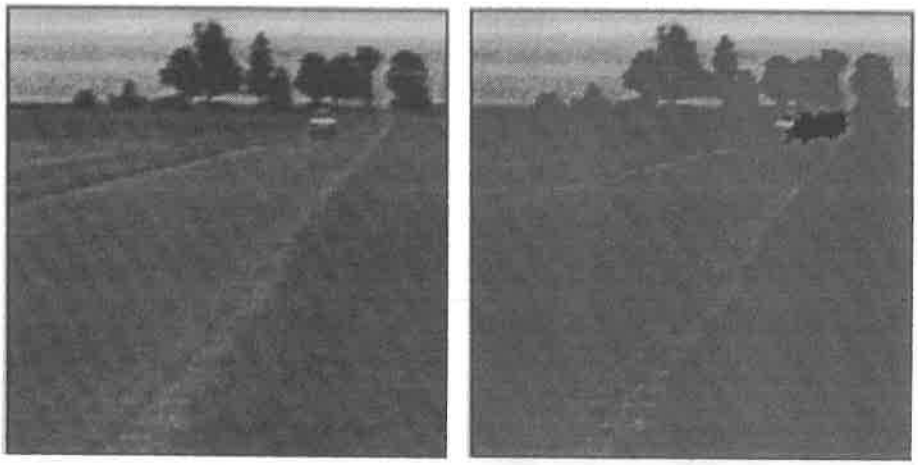
(g) (h)

Fig. 4. (*Continued.*) Frames 60 and 80 are shown in (e) and (g), respectively. Corresponding time projections are shown in (f) and (h).

(a) (b) (c)

Fig. 5. A 140-frame road scene with an approaching car, digitized from a video camera. Frames 31 and 51 are shown in (a) and (c), respectively. Corresponding time projections of the results are shown in (b) and (d). Detected trajectories are shown in black, superimposed over the original images. A 17-stage test is used with discrete test set 17s0-0.3, Table IV. *(Continued on next page.)*

(d) (e) (f)

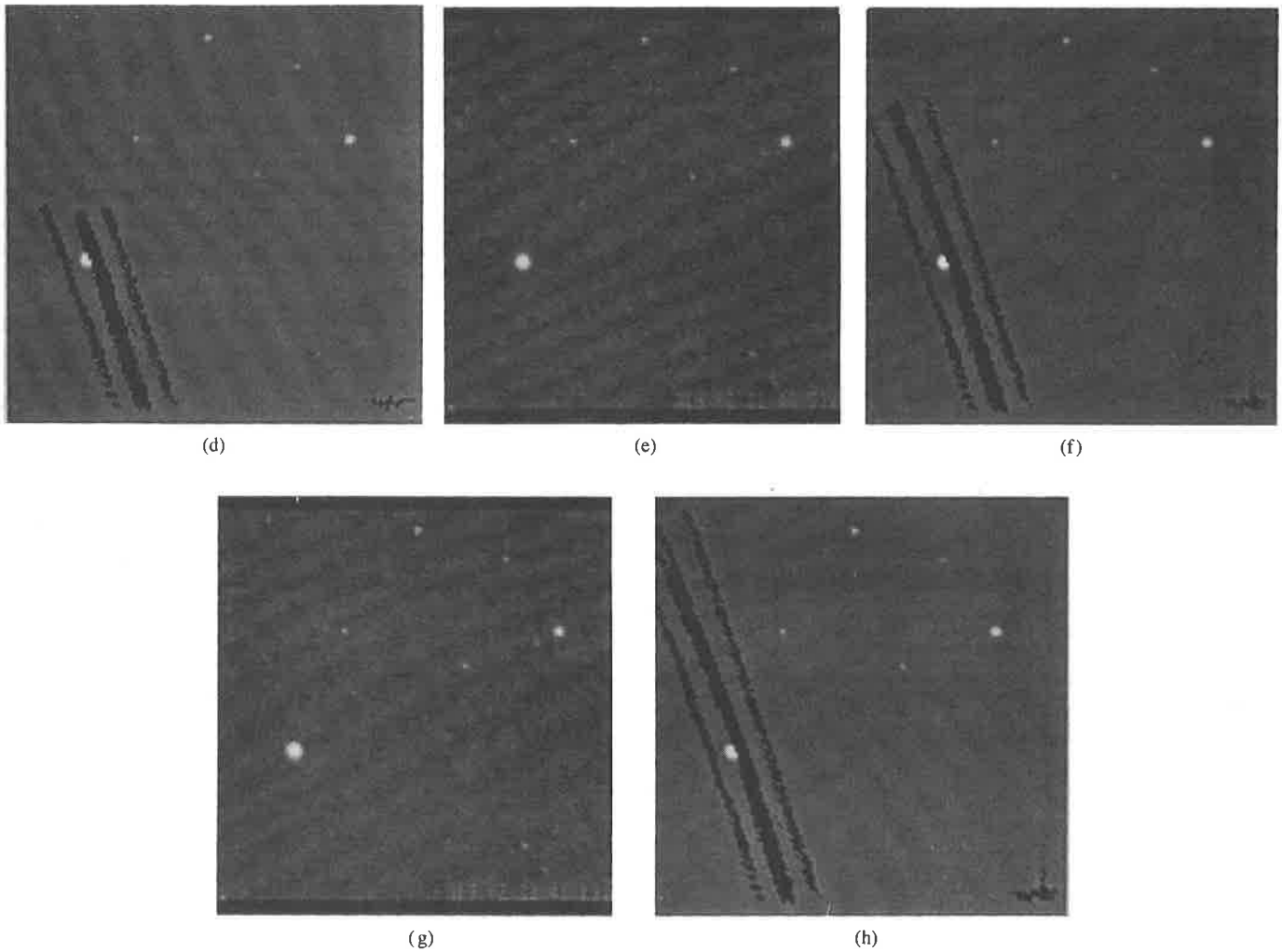


(g) (h)

Fig. 5. (*Continued.*) Frames 71 and 91 are shown in (e) and (g), respectively. Corresponding time projections are shown in (f) and (h).

(a) (b) (c)

Fig. 6. A sequence of 78 images of the night sky, digitized by a CCD array at the output of a telescope. Frames 20 and 40 are shown in (a) and (c), respectively. Corresponding time projections of the results are shown in (b) and (d). Detected trajectories are shown in black, superimposed over the original images. A 17-stage test is used with discrete test set 10s0-2, Table IV.
(Continued on next page.)



(d) (e) (f)

(g) (h)

Fig. 6. (*Continued*.) Frames 60 and 78 are shown in (e) and (g), respectively. Corresponding time projections of the results are shown in (f) and (h).