Statistical Sequential Analysis for Real-Time Video Scene Change Detection on Compressed Multimedia Bitstream

Dan Lelescu and Dan Schonfeld

***Abstract*—The increased availability and usage of multimedia information have created a critical need for efficient multimedia processing algorithms. These algorithms must offer capabilities related to browsing, indexing, and retrieval of relevant data. A crucial step in multimedia processing is that of reliable video segmentation into visually coherent video shots through scene change detection. Video segmentation enables subsequent processing operations on video shots, such as video indexing, semantic representation, or tracking of selected video information. Since video sequences generally contain both abrupt and gradual scene changes, video segmentation algorithms must be able to detect a large variety of changes. While existing algorithms perform relatively well for detecting abrupt transitions (video cuts), reliable detection of gradual changes is much more difficult. In this paper, a novel one-pass, real-time approach to video scene change detection based on statistical sequential analysis and operating on compressed multimedia bitstream is proposed. Our approach models video sequences as stochastic processes, with scene changes being reflected by changes in the characteristics (parameters) of the process. Statistical sequential analysis is used to provide an unified framework for the detection of both abrupt and gradual scene changes.**

***Index Terms*—Abrupt and gradual scene changes, scene change detection, video segmentation, video shots.**

I. Introduction

GIVEN the rapid expansion in the volume of multimedia data and the increasing demand for fast access to relevant data, the need for parsing, interpretation, and retrieval of multimedia information has become of paramount importance. Algorithms designed for multimedia processing must offer capabilities such as browsing, indexing, tracking, and retrieval of relevant information. Since the original data are compressed to preserve storage space, and decompression is extremely costly, the algorithms designed to operate with multimedia data should ideally do so in the compressed (or minimally-decoded) domain. An important step in processing multimedia information is the parsing of video into relatively visually-coherent segments called *video shots*, delimited by scene changes. Video segmentation is needed to enable subsequent processing operations on video shots, such as video indexing, semantic representation, or tracking of selected video information. After segmentation into shots, video indexing can be performed to facilitate a compact representation of information in each video shot, to be used for later retrieval. Algorithms can also perform video analysis or retrieval by tracking specific objects in video shots, as it was presented for example in [1]. In this paper, we propose a new real-time approach for detecting scene changes based on statistical sequential analysis, and operating on minimally-decoded video bitstreams. The algorithms developed were tested on videos encoded using the MPEG-2 video compression standard. However, they can be easily extended for use with different video compression methods.

In general, video sequences contain two main types of scene changes: abrupt and gradual. With the increasing use of special effects as a method of transition from one video shot to another, it should be assumed that video sequences contain both abrupt and different types of gradual changes. Thus, the algorithms for scene change detection should not be designed only for detection of abrupt transitions, or specific gradual changes such as fades or dissolves. A challenge in video segmentation is that the detection algorithm must be able to handle all types of scene changes. Whereas abrupt changes (cuts) are relatively easy to detect, gradual changes present more difficulty for detection. There is a large variety of gradual transitions that can be produced. A gradual scene change usually takes place over a variable number of frames (having different rates of change). Generally, the video change from frame to frame is small. Thus, it is difficult to determine directly in the compressed domain that a gradual change is taking place without considering a sufficiently large number of pictures. It is also desirable to formalize the analysis of video scene changes beyond the use of empirical observations about the nature of the transitions, which may or may not hold in all cases.

In order to address these challenges, we pose the scene change detection problem in the context of statistical sequential representation and analysis of the video information. The nature and volume of video data, and the characteristics of video scene changes are well-suited to a stochastic modeling. Thus, a video sequence is considered to be a realization of a stochastic process. Scene changes are considered to be represented by changes in the characteristics (parameters) of the stochastic process. We use statistical analysis to monitor parameters of the stochastic process, to allow the design of an

Manuscript received March 28, 2000; revised December 18, 2001. This work was performed while the first author was with the University of Illinois-Chicago. The associate editor coordinating the review of this paper and approving it for publication was Dr. Tat-Seng Chua.

D. Lelescu is with DoCoMo Communications Laboratories, San Jose, CA 95110 USA (e-mail: lelescu@docomolabs-usa.com).

D. Schonfeld is with the Multimedia Communications Laboratory, Department of Electrical and Computer Engineering University of Illinois-Chicago, Chicago, IL 60607-7053 USA (e-mail: ds@ece.uic.edu).

Digital Object Identifier 10.1109/TMM.2003.808819

unified approach to scene change detection for various types of changes (both abrupt and gradual). Most special effects have particular characteristics that make them hard to detect with deterministic methods, usage of simple statistics (e.g., intensity variances, histograms), and considering only a small number of pictures at a time, which has been typically the case in existing algorithms. In our approach, through use of *dimensionality reduction* and *sufficient statistics*, a large enough number of pictures can be analyzed for gradual scene change detection. The dimensionality reduction is achieved by utilizing an efficient implementation of an optimal transformation (principal component analysis). The sufficient statistic used is based on the concept of generalized likelihood ratio. Thus, the main steps used for detecting video scene changes are the dimensionality reduction of the original video data followed by change detection on the resulting feature vectors.

The use of the statistical sequential analysis facilitates a unified approach for detecting both abrupt and gradual transitions, since any scene change is modeled as a change in parameters of the stochastic process. Thus, the dynamics of the video data dictates when a scene change is detected. The evidence of a scene change as reflected by changes in a sequentially computed test statistic is used for detection. No *a priori* assumptions must be made about the specific model of the scene change (e.g., linear). No predefined time window needs to be used for analyzing the data to detect gradual scene changes, since theoretically, all the samples from the beginning of the current video shot can be represented in the test statistic. Thus, variable-length gradual changes can be detected, by allowing the statistical data pertaining to a scene change to accumulate and trigger a detection. The sequential aspect of the approach enables the one-pass detection of scene changes.

The resulting algorithms use one threshold applied to the test statistic that responds to scene changes, for the purpose of isolating the transitions. Therefore, the disadvantage of using difficult to adjust combinations of multiple thresholds found in some existing scene change detection approaches is eliminated. The novel scene change detection approach presented in this paper offers a one-pass, real time solution to detecting both abrupt and gradual changes by operating on minimally-decoded video bitstreams. To illustrate this approach, additive and nonadditive models of video scene changes and the performance of the corresponding detection algorithms are discussed in the context of different parameterizations.

This paper is organized as follows. Section II covers related work in the area of video segmentation. In Section III, the dimensionality reduction and representation of the original video data is discussed. The statistical scene change detection approach is discussed in Section IV. Section V contains simulation results. The paper is concluded in Section VI.

II. Related Work

Algorithms for scene change detection can be broadly classified as operating on decompressed data, or working directly with the compressed information, using minimally-decoded data such as macroblock (MB) types, motion vectors, or DC-coefficient images. The original video information is compressed to preserve storage space and decompression of the data, in particular the inverse transform operation (inverse Discrete Cosine Transform), is very costly. There have been a number of video segmentation approaches that appeared in the literature [2]–[6], which operate on the decompressed data, in the pixel domain. They may use methods based on color histogram differences [2], changes in the edge characteristics in the image [3], characteristic patterns in the standard deviation of pixel intensities for detection of fades [5], or linear modeling of transitions in gradual changes [4]. The color histogram-based methods are the most reliable for detection of abrupt changes. In the twin comparison algorithm in [2], it is observed that during a dissolve, successive picture histogram differences are greater than the average within-shot histogram difference, but each of them is not large enough to be classified as a change by itself. Thus, two thresholds for change detection can be used, one lower and one higher. The higher threshold is intended to detect the abrupt transitions. To detect dissolves, the accumulated difference over a range of pictures with successive histogram differences greater than the lower threshold, but not large enough to individually exceed the larger threshold, should exceed the larger threshold. A good comparative assessment of some of these approaches can be found in [6] and [7]. The comparison in [6] takes into account newer algorithms designed to detect more complex edits (fades and dissolves). The author finds that the performance of change detectors based on the edge change ratio (ECR) [3] was inferior to that of specialized detectors based on color histogram differences [8], standard deviation of pixel intensities and edge-based contrast. Many dissolves do not show the characteristic edge pattern described in [3], especially the long dissolves where the ECR change is so slight that it is hidden in noise. An edge based method for cut detection is also presented in [9]. In addition to operating on decompressed data, a common feature of these approaches is that they work relatively well on abrupt changes or very specific transitions such as fades, but fail on the detection of general gradual changes. Also, analysis of a limited number of pictures at a given time is detrimental to the algorithm's capacity to handle the diverse types of gradual changes. This is true for algorithms that work in both the decompressed and compressed domain.

In an approach that can detect both abrupt and gradual scene changes [10], the authors observe that a video shot boundary can be construed as a temporal multiresolution edge event in a feature space. The feature space can be obtained by considering color, shape, texture, etc. A color histogram is adopted to represent each video frame. A wavelet technique is applied to the video frames in this representation space, and a multiple resolution analysis is used to identify both abrupt and gradual scene changes. Different transitions with different durations are visible at different resolutions. Viewing the wavelet transform as a convolution operation, the lengths of the wavelet filter varies between two and 100 frames. The authors report a Recall of 98.5%, Precision of 77.9% for video cuts, and a Recall of 97.1% and Precision 65.2% for gradual transitions.

Recently, the emphasis in multimedia processing has been placed on algorithms operating in the compressed domain [11]–[16]. Our proposed approach falls into this category.

Scene change detection algorithms operating on compressed data may use MB and motion type information for detecting transitions [11], [12], [16]. In the case of gradual changes or special effects, this information may not be sufficient for detecting the change (especially if only a few pictures are analyzed at a given time). For example, for abrupt cuts occurring in P pictures, it is expected that macroblocks will be mostly intracoded since they cannot be predicted well from the different reference picture information. Similarly, for a cut appearing in a B picture, it is assumed that macroblocks will be either intracoded or backward-predicted (to take advantage of a future reference picture that contains similar information). Depending on the difference between the visual information in the old and new video shot, these conditions may have various degrees of reliability. In the case of gradual changes, the problem is compounded by the fact that MBs can remain bidirectionally-predicted in B pictures, or predicted (not intracoded) for P pictures.

Similarly to our approach, at the level of preprocessing for operation on minimally-decoded video data, DC coefficients (reduced DC images) are utilized in [12]–[15] for scene change detection. In [13], DC coefficients of pictures that are at some distance from each other temporally are used to construct a comparison metric based on the sum of absolute differences. In [15] DC histograms are used to detect scene changes. Detection of dissolves is considered using averages of past histograms. In [12], the differences in the luminance and chrominance DC histograms between I pictures are utilized for change detection, along with MB motion type information for the predicted (P,B) pictures. Also, the specific shape of the intensity variance of DC coefficients in I and P pictures is detected as an indication of a particular type of gradual transition (dissolve). As with corresponding histogram-based algorithms working in the pixel domain, detection algorithms that use DC histograms averages or accumulated differences in a time window have a number of disadvantages. These are related to the fact that gradual changes have widely-variable length which makes it difficult to set the size of the time window *a priori*, and the algorithms may use multiple thresholds that must be adapted for each video shot.

In [17], a comparison of metrics used for scene change detection is performed. The metrics are applied to two successive images and their response to a scene change is recorded. These metrics include chi-square test of histograms, absolute values of histogram differences, likelihood ratio, Snedecor's F-test, template matching, inner product, and three newly-introduced metrics. The metrics were tested on 416 frames. There were 340 pairs with no changes and 75 pairs with changes. The F-test and the three new metrics performed the best overall for abrupt cuts, but required more computation time. Using a test sample of six fades and seven dissolves, it was found that the statistics based metrics performed well for this case. In [19], which is based on the authors earlier work [18], the chi-square test using intensity histograms of two successive I-frames of an MPEG video is utilized for scene change detection. Global, and row and column histograms are used in the decision process. Since a gradual transition spans several frames and can cause a detection decision spanning two or more I-frames, the system waits for the next I-frame (latency) before outputting the decision for the current I-frame. The assumption is that the distance between the I-frames is 12. The ground truth for the scene changes included the motion-induced transitions.

In [14], DC coefficients representing each image are mapped to a lower-dimensional space using the FastMap transform. In our approach, the lower-dimensional representation of the DC coefficients for each picture is based on the Karhunen–Loeve transformation. After the lower dimensional representation of the original data we use stochastic modeling and analysis for change detection. In [14], in the lower dimensional space, the video sequence is represented by a trail of points (called *VideoTrail*). By clustering and analyzing the geometrical distribution of points in the space, scene changes are detected as sparse trails. The performance and complexity of this method are dependent on the clustering and geometrical analysis algorithms used in a multidimensional space. In [20], a comparison between the authors' VideoTrails (VT) approach and various existing algorithms for scene change detection is presented. There are three versions of the VT algorithm using reduced ten-dimensional YUV input data. These are compared to the Plateau [13], VarCurve [12], and Twin Comparison [2] algorithms. Natural video simulations were performed, with special effects that range from short duration (five frames) to more than 100 frames in length. With global motion or camera motion alarm removal, the VT-based algorithms performed the best. The best overall VT-based algorithm produced a Recall of 62.3%, and Precision 69.7%. Another VT version yielded a higher Recall of 68% with a lower Precision of 57.1%. The Recall and Precision for the Twin Comparison algorithm were $R = 61.2\%,\ P = 53.7\%$.

In addition to differences further discussed later, in our simulations we do not include the gradual motion-related alarms in the ground truth, unless they are so abrupt that a histogram-based detector is also detecting them. Instead, they are counted as false alarms to account for the worst-case scenario when a global motion detector is not available. The algorithms introduced in this paper do not utilize a global motion detector, which is capable of marking these motion-induced scene changes depending on the application domain.

Our approach departs from the concept of fixed-size sample change detection, and successive deterministic or statistical comparisons between pairs of frames to detect scene changes. In this paper, we propose an approach for scene change detection that is based on the more powerful concept of open statistical tests, which assumes a sequential decision process based on theoretically the entire past. The sequential aspect of the approach is critical for a real-time, one-pass solution where we do not have the entire sample (sequence) at our disposal, rather we are receiving one sample at the time. In conjunction with the previous considerations, the model used for the video sequences is that of a stochastic process where scene changes are reflected by changes in the characteristics (parameters) of the process. This allows for flexibility in detecting various types of changes without the constraints of imposing a specific model of the change (e.g., linear). No predefined comparison distance is set, such as for example K for comparing frame i and $i+K$. No limited time window is set and used, extending in the past and/or future (no latency), in order to make a decision.

Also, with the exception of very abrupt changes (cuts), considering only a few pictures for scene change detection analysis is not reliable. For gradual scene changes, where the video information varies slowly from picture to picture, the statistical information about the change is allowed to accumulate and trigger the detection. This emphasizes an unified approach to detecting different types of changes having variable length.

III. Dimensionality Reduction and Feature Vector Representation

The preprocessing phase prior to applying the change detection consists of dimensionality reduction and a lower-dimensional representation of the original video data. The first step to reduce dimensionality is to consider only the DC coefficients of each block in the image. For efficiency and practical purposes, only I- and P-pictures are utilized by the algorithm. Luminance and chrominance (YUV) DC coefficients are considered for each macroblock in the image. The DC coefficients are readily available in I-pictures, which are intracoded. A widely-used method for estimating the DC coefficients in P-pictures is presented in [21]. Thus, a matrix of DC coefficients—the DC image—is obtained for each picture. For the luminance and the chrominance components respectively (YUV), a vector of DC coefficients is formed by lexicographic ordering, and the three resulting vectors are merged into a *DC vector* corresponding to each picture.

In general, and more so if a real-time constraint is imposed on the process of scene change detection, some degree of dimensionality reduction of the original DC image data can be performed. Dimension reduction similar to ours has also been recently used by other approaches in the literature, for the purpose of scene change detection (see for example [14], where the FastMap transform was used to reduce the dimensionality of the original DC image space to $K = 10$). In this paper, the transformation chosen for this purpose is the principal component analysis (PCA). We model the video sequence, with each picture represented by its DC vector, as a stochastic process. The Karhunen–Loeve transformation (KLT) gives the optimal representation of a stochastic process in a finite-dimensional vector space, in the MSE sense. PCA performs a partial KLT by identifying the largest eigenvalues and retaining only the corresponding eigenvectors. The PCA transform retains the most important features of the images. We should note that the scene change detection approach presented in this paper is not constrained to work with a specific type of representation of the video data. The input can be represented by the sequence of original DC images (vectors), or vectors corresponding to the histogram bins of the images.

We consider a training set of DC data vectors corresponding to images in the beginning of each video shot for the purpose of subspace determination. Let $\mathbf{P}$ be an $N_c \times (M+1)$ *data matrix*, whose columns are the DC vectors from the training set. N_c is the dimension of the DC-coefficient data vectors and $M+1$ is the size of the training set (number of vectors considered). The PCA finds eigenvectors of the estimated correlation matrix $\mathbf{C} = \mathbf{PP^T}$, which is an $N_c \times N_c$ matrix. The size of matrix $\mathbf{C}$ is generally very large, which would result in computationally intensive operations. As described in [22], an efficient approach for negotiating this problem is to consider the implicit matrix $\tilde{\mathbf{C}} = \mathbf{P^T P}$. The matrix $\tilde{\mathbf{C}}$ is of size $(M+1) \times (M+1)$, which is much smaller than the size of $\mathbf{C}$. The M *largest* eigenvalues λ_i, and corresponding eigenvectors e_i of $\mathbf{C}$ can be found from the M largest eigenvalues and eigenvectors of $\tilde{\mathbf{C}}$ as follows [22]:

$$\begin{aligned} \lambda_i &= \tilde{\lambda}_i \\ e_i &= \tilde{\lambda}_i^{-(1/2)} \mathbf{P} \tilde{e}_i \end{aligned} \tag{1}$$

where $\tilde{\lambda}_i, \tilde{e}_i$ are the corresponding eigenvalues and eigenvectors of $\tilde{\mathbf{C}}$.

The $N_c \times M$ eigenmatrix $\mathbf{\Phi}$ is formed, having as columns the principal eigenvectors e_i. Using the matrix $\mathbf{\Phi}$, each original DC vector sample X_k in the current video shot, can be represented using a reduced-dimensionality $M \times 1$ *feature vector*, $Y_k = \mathbf{\Phi}^T X_k$, where k is the time index. Feature vectors represent the input to the change detection algorithm.

IV. Scene Change Detection

An extensive literature exists related to the area of statistical sequential analysis and in particular change detection theory (see [23] and references therein). The video scene change detection approach presented in this paper is based on the change detection theory. The dimensionality reduction for DC data vectors X_k corresponding to each I or P picture, is performed as described in Section III. Thus, a sequence of corresponding feature vectors Y_k is obtained. The change detection algorithm operates sequentially on the feature vectors. The problem of scene change detection becomes one of detecting changes in the parameters of the probability density function (pdf) associated with this vector random process.

A. Additive Modeling

Let us first discuss the case where a scene change is modeled as an additive change in the mean parameter of the pdf associated with the feature vector sequence that describes the video data. The reduced dimensionality feature vectors, corresponding to I- and P-pictures, are assumed to form an i.i.d. sequence of r-dimensional random vectors $\{Y_k\}$ having Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, with pdf:

$$p_{\mu,\Sigma}(Y_k) = \frac{1}{\sqrt{(2\pi)^r (\det\Sigma)}} e^{-(1/2)(Y_k-\mu)^T \Sigma^{-1} (Y_k-\mu)}. \tag{2}$$

The scene change is modeled as a change in the vector parameter $\theta = \mu$ of the pdf characterizing the feature vector random process. The parameter $\theta = \theta_0$ before the change, and $\theta = \theta_1$ after the change. Depending on the amount of information available about the parameter before and after the change, the problem formulation is correspondingly different. Given the requirement of operating in real-time, in general minimal or no information is available about the parameter $\theta = \theta_1$ after the change. Let us begin by formulating the problem of testing between the following two composite hypothesis:

$$\begin{aligned}\mathbf{H_0} &= \{\theta : \|\theta - \theta_0\|^2_\Sigma \leq a^2, k < t_0\}\\ \mathbf{H_1} &= \{\theta : \|\theta - \theta_0\|^2_\Sigma \geq b^2, k \geq t_0\}\end{aligned} \tag{3}$$

where $\|\theta - \theta_0\|^2_\Sigma = (\theta - \theta_0)^T \Sigma^{-1} (\theta - \theta_0)$, t_0 is the true change time and $a < b$. Therefore, the formulation of the hypothesis testing problem reflects the case where there is a known upper bound for θ_0 and a known lower bound for θ_1. The case of interest where θ_0 is assumed to be known, and θ_1 is assumed completely unknown, can be obtained as a limit case of the solution to the problem above, as we shall see below.

The generalized likelihood ratio (GLR) [24] algorithm that gives the solution to the hypothesis testing problem in (3), uses a *generalized likelihood ratio*, where the unknown parameters are replaced by their maximum likelihood (ML) estimates. Thus, for the problem formulation in (3), the generalized likelihood ratio for a sequence of vectors $\{Y_j, \ldots, Y_k\}$ is

$$S_j^k = \ln \frac{\sup_{\|\theta - \theta_0\|_\Sigma \geq b} p_\theta(Y_j, \ldots, Y_k)}{\sup_{\|\theta - \theta_0\|_\Sigma \leq a} p_\theta(Y_j, \ldots, Y_k)} \tag{4}$$

where p_θ is the corresponding parameterized probability density function. The sequential GLR algorithm is then

$$\begin{aligned}t_a &= \min\{k \geq 1 : g_k \geq h\}\\ g_k &= \max_{1 \leq j \leq k} S_j^k\end{aligned} \tag{5}$$

where k is the discrete time index, t_a is the alarm (detection) time, g_k is the *test statistic*, and h is a threshold. Thus, the test statistic is based upon the likelihood ratio, which can be easily proven to be a sufficient statistic.

Under the assumption of an i.i.d. sequence, S_j^k becomes

$$S_j^k = \ln \frac{\sup_{\|\theta - \theta_0\|_\Sigma \geq b} \prod_{i=j}^k p_\theta(Y_i)}{\sup_{\|\theta - \theta_0\|_\Sigma \leq a} \prod_{i=j}^k p_\theta(Y_i)}. \tag{6}$$

Given the i.i.d. Gaussian assumption, S_j^k can be written as [see (2)]

$$\begin{aligned}S_j^k &= \ln \frac{\sup_{\|\theta - \theta_0\|_\Sigma \geq b} e^{-(1/2)\sum_{i=j}^k (Y_i - \theta)^T \Sigma^{-1} (Y_i - \theta)}}{\sup_{\|\theta - \theta_0\|_\Sigma \leq a} e^{-(1/2)\sum_{i=j}^k (Y_i - \theta)^T \Sigma^{-1} (Y_i - \theta)}}\\ &= \sup_{\|\theta - \theta_0\|_\Sigma \geq b} \left\{ -\frac{1}{2} \sum_{i=j}^k (Y_i - \theta)^T \Sigma^{-1} (Y_i - \theta) \right\}\\ &\quad - \sup_{\|\theta - \theta_0\|_\Sigma \leq a} \left\{ -\frac{1}{2} \sum_{i=j}^k (Y_i - \theta)^T \Sigma^{-1} (Y_i - \theta) \right\}.\end{aligned} \tag{7}$$

As presented in [24], S_j^k is obtained:

$$K_l S_j^k = \begin{cases} -(\chi_j^k - b)^2, & \chi_j^k < a\\ -(\chi_j^k - b)^2 + (\chi_j^k - a)^2, & a \leq \chi_j^k \leq b\\ +(\chi_j^k - a)^2, & \chi_j^k > b \end{cases} \tag{8}$$

where $K_l = 2/(k - j + 1)$ and

$$\begin{aligned}\chi_j^k &= \left[(\bar{Y}_j^k - \theta_0)^T \Sigma^{-1} (\bar{Y}_j^k - \theta_0)\right]^{1/2}\\ \bar{Y}_j^k &= \frac{1}{k - j + 1} \sum_{i=j}^k Y_i.\end{aligned} \tag{9}$$

Thus, for the current problem formulation, at each time index k, the GLR algorithm in (5) uses S_j^k as given by (8). The data that are needed in (9) are the feature vectors Y_i, the covariance Σ, and $\theta_0 = \mu_0$, all variables that can be determined as detailed later. The expression for $\bar{Y}_j^k$ admits an iterative computation, allowing for a fast operation of the detection algorithm. Only updates are computed at each time k, with the arrival of a new data vector Y_k. As a practical implication, the feature vectors Y_k corresponding to the pictures do not have to be memorized, and no permanent processing buffer is needed.

For the case of interest, the parameter before the change, θ_0, is assumed to be known and the parameter after the change is assumed completely unknown but different than θ_0. This is a special case of the hypothesis testing problem in (3), which can be written as follows:

$$\begin{aligned}\mathbf{H_0} &= \{\theta : \theta = \theta_0, k < t_0\}\\ \mathbf{H_1} &= \{\theta : \theta \neq \theta_0, k \geq t_0\}.\end{aligned} \tag{10}$$

Thus, we let $a = b = 0$ (the limit case) in (3) and (8). Therefore, the GLR algorithm in (5) becomes

$$\begin{aligned}t_a &= \min\{k \geq 1 : g_k \geq h\}\\ g_k &= \max_{1 \leq j \leq k} \left\{ \frac{k - j + 1}{2} (\chi_j^k)^2 \right\}\end{aligned} \tag{11}$$

with χ_j^k as in (9). In the development above, θ_0 is assumed to be known. In fact, θ_0 can be estimated using a number M of feature vectors in the beginning of each video shot. The covariance Σ is estimated using the same set of M vectors. Also, theoretically, at each k, the statistic is constructed based on all data from $1 \ldots k$ (the entire past, see (11)). In practice, we can readily consider a sufficiently large maximum number of past data points L, that begins at $[k - (L - 1)]$ and ends at k.

Let us now summarize the overall strategy for detecting additive changes in the mean parameter of the pdf describing the video sequence, for the case of multiple scene changes. The first M original DC vectors, $\{X_k\}$, in each shot, are used to determine the subspace representation using the PCA (see Section III). Thus, each original $N_c \times 1$ DC data vector X_k, corresponding to I- and P-pictures in the video sequence, can now be transformed into reduced-dimensionality feature vectors Y_k. The first M of these vectors in each video shot are used to estimate the covariance Σ, and the mean parameter $\theta_0 = \mu_0$. Sequentially, feature vectors $\{Y_k\}$ are the input to the change detection algorithm. The change detection algorithm is activated at the $(M + 1)$th I- or P-picture in a video shot, and it continues operating sequentially until a scene change is detected. The process described above is repeated for each new video shot and is depicted in Fig. 1.

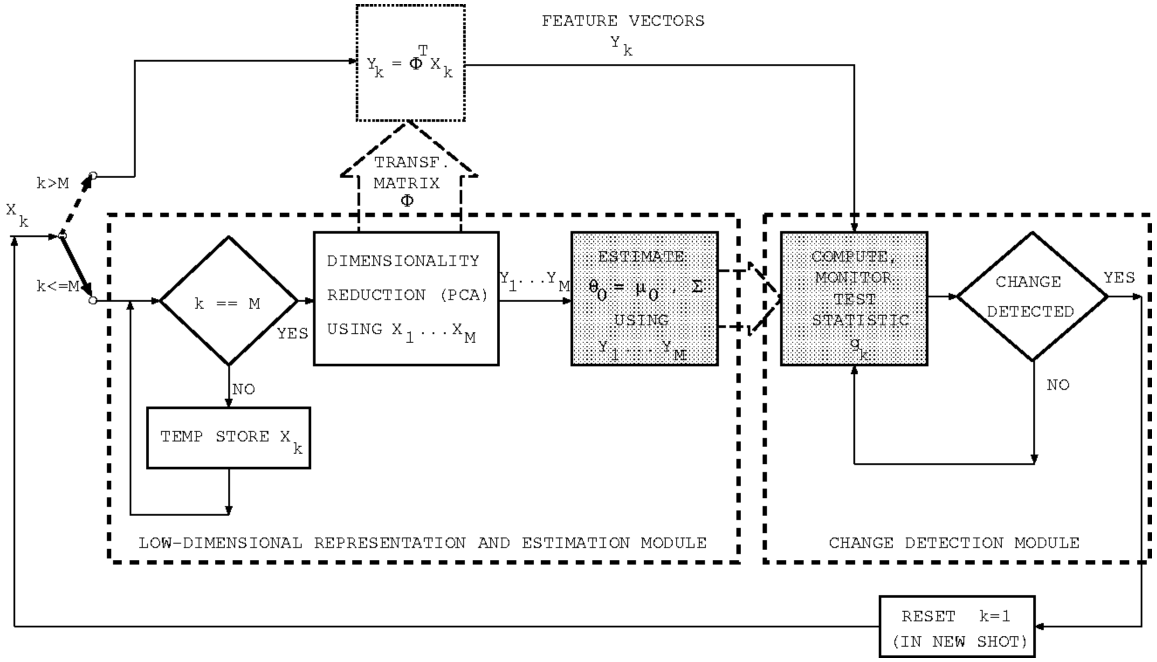

FEATURE VECTORS
Y_k
$Y_k = \Phi^T X_k$
TRANSF. MATRIX
Φ
k>M
X_k
k<=M
k == M
YES
NO
DIMENSIONALITY REDUCTION (PCA) USING $X_1 \ldots X_M$
$Y_1 \ldots Y_M$
ESTIMATE $\theta_0 = \mu_0$, Σ USING $Y_1 \ldots Y_M$
COMPUTE, MONITOR TEST STATISTIC g_k
CHANGE DETECTED
YES
NO
TEMP STORE X_k
LOW-DIMENSIONAL REPRESENTATION AND ESTIMATION MODULE
CHANGE DETECTION MODULE
RESET k=1 (IN NEW SHOT)

Fig. 1. Block diagram for scene change detection.

The shaded blocks indicate the points where the currently discussed additive model scene change detector differs computationally from the nonadditive detector which will be presented in Section IV-B. A pseudocode that summarizes the main steps of the scene change detection algorithm is also provided as follows.

```
01: while(!EOF) {/* End of video sequence not reached */
02:  bSceneChange = FALSE;
03:  $k = 1$; /* Reset $k$ for a new video shot */
04:  while ($k <= M + 1$) {/* initialization phase */
05:    get $X_k$;
06:    temp store $X_k$;
07:    if ($k == M + 1$){
08:      compute PCA, $\Phi$, and $Y_1, \ldots, Y_M$;
09:      estimate $\theta_0$, using $Y_1, \ldots, Y_M$
10:    }
11:    else
12:      $k = k + 1$;
13:  }
14:  while(!bSceneChange) {/* Starting with $X_{M+1}$ in current shot,change detection */
15:    get $X_k$;
16:    $Y_k = \Phi^T X_k$;
17:    compute $g_k$; /* (11) */
18:    if ($g_k < h$)
19:      $k = k + 1$;
20:    else {/* Scene change detected */
21:      bSceneChange = TRUE;
22:      alarm(); /* Result: go to statement 01, new video shot */
23:    }
24:  }
25: }
```

B. Non-Additive Modeling

While maintaining the general model of a video sequence as a vector random process, in this section we investigate the effect of different assumptions about the model and about the nature of the changes in the video sequence. As before, we shall assume that the vector random process $\{Y_k\}$ is an i.i.d. sequence that can be described with a model that uses a parameterized probability density function p_θ. The objective is to detect changes in the vector parameter $\theta \in \mathbb{R}^r$. For nonadditive changes the likelihood ratio may become computationally complex. One way to address this problem is to consider using the *asymptotic local hypotheses approach* for designing the change detection algorithm (as in [24]). A brief required background follows.

Let us consider a parametric family of distributions $\mathcal{P} = \{P_\theta\}_{\theta \in \Psi}$ with $\Psi \subset \mathbb{R}^r$, and a sample of size N[24]. Assume there exists a convergent sequence of points in $\mathbb{R}^r$ denoted by $\nu_N \Upsilon$, such that $\nu_N \to \nu \in \mathbb{R}$, where, without loss of generality, Υ is the vector of change direction with $||\Upsilon|| = 1$. Let us use the notation $\theta_N = \theta + (\nu_N/\sqrt{N})\Upsilon$. Then, the two distributions P_θ, P_{θ_N}, corresponding to the two hypotheses

$$\begin{aligned} \mathbf{H_0} &= \{\{Y_k\} \sim P_\theta\} \\ \mathbf{H_1} &= \left\{\{Y_k\} \sim P_{\theta_N = \theta + (\nu_N/\sqrt{N})\Upsilon}\right\} \end{aligned} \quad (12)$$

get closer to each other when $N \to \infty$. Let us write the log–likelihood ratio for the sample $\mathcal{Y}_1^N = \{Y_1, \ldots, Y_N\}$ as:

$$S(\theta, \theta_N) = \ln \frac{p_{\theta_N}(\mathcal{Y}_1^N)}{p_\theta(\mathcal{Y}_1^N)}. \tag{13}$$

Definition [24]: The parametric family of distributions $\mathcal{P} = \{P_\theta\}_{\theta \in \Psi}$ is called *locally asymptotic normal (LAN)* if the log-likelihood ratio for hypotheses $\mathbf{H_0}$ and $\mathbf{H_1}$ can be written as

$$S(\theta, \theta_N) = \nu \Upsilon^T \Delta_N(\theta) - \frac{\nu^2}{2} \Upsilon^T I_N(\theta) \Upsilon + \alpha_N(\mathcal{Y}_1^N, \theta, \nu\Upsilon) \tag{14}$$

where the vector efficient score

$$\Delta_N(\theta) = \frac{1}{\sqrt{N}} \frac{\partial \ln p_\theta(\mathcal{Y}_1^N)}{\partial \theta} = \frac{1}{\sqrt{N}} \mathcal{Z}_N \tag{15}$$

$I_N(\theta)$ is the Fisher information matrix (covariance of $\Delta_N(\theta)$), and the following asymptotic normality holds:

$$\Delta_N(\theta) \sim \mathcal{N}(0, I(\theta)). \tag{16}$$

The random variable α_N is such that $\alpha_N \to 0$ almost surely under the probability measure $P_{\theta + 1/\sqrt{N}\nu\Upsilon}$. A corollary of the LAN properties for parametric family $\mathcal{P}$ satisfying regularity conditions, is that the vector efficient score $\Delta_N(\theta)$ is an asymptotic sufficient statistic.

Considering the sequential context and the parameterized p.d.f. p_θ describing the data, a nonadditive change in the vector parameter θ is to be detected, under the assumption that the parameter before the change θ_0 is known (estimated), and the parameter after the change θ is unknown, and different than θ_0. In this case, the two hypotheses are formulated in the context of the local asymptotic assumption, for a change around a reference parameter θ_0, to θ. Thus, the two hypotheses are [24]

$$\begin{aligned} \mathbf{H_0} &= \{\theta : \theta = \theta_0, k < t_0\} \\ \mathbf{H_1} &= \{\theta : \|\theta - \theta_0\| = \mu, k \geq t_0\} \end{aligned} \tag{17}$$

where t_0 is the change time for the parameter, k is the temporal index, and $\mu > 0$ is small.

The Generalized Likelihood Ratio algorithm is the relevant method for solving this hypothesis testing problem. Thus, for the sequential case, the GLR algorithm can be written as

$$\begin{aligned} t_a &= \min\{k \geq 1 : g_k \geq h\} \\ g_k &= \max_{1 \leq j \leq k} \sup_\theta S_j^k(\theta_0, \theta). \end{aligned} \tag{18}$$

The log-likelihood ratio $S_j^k(\theta_0, \theta)$ for the sequence of vectors $\mathcal{Y}_j^k = \{Y_j, \ldots, Y_k\}$, $j \leq k$ is

$$S_j^k(\theta_0, \theta) = \ln \frac{p_\theta(\mathcal{Y}_j^k)}{p_{\theta_0}(\mathcal{Y}_j^k)}. \tag{19}$$

In the context of the local asymptotic approach, similarly to (14), the second-order Taylor expansion of the log-likelihood ratio around θ_0, for the sample $\mathcal{Y}_j^k = \{Y_j, \ldots, Y_k\}$ is [24]:

$$S_j^k(\theta_0, \theta) \approx (\theta - \theta_0)^T \mathcal{Z}_j^k(\theta_0) - \frac{1}{2}(\theta - \theta_0)^T I(\theta_0)(\theta - \theta_0) \tag{20}$$

where $I(\theta)$ is the Fisher information matrix, and $\mathcal{Z}_j^k$ is the efficient score for sample $\mathcal{Y}_j^k = \{Y_j, \ldots, Y_k\}$ given by

$$\mathcal{Z}_j^k = \frac{\partial \ln p_\theta(\mathcal{Y}_j^k)}{\partial \theta}. \tag{21}$$

Using the second-order expansion (20) and the i.i.d assumption for the sample $\mathcal{Y}_j^k$ of size $(k - j + 1)$, we have [24]

$$\sup_\theta S_j^k(\theta_0, \theta) \approx \frac{k - j + 1}{2} \left(\bar{Z}_j^k\right)^T I^{-1} \left(\theta_0\right) (\bar{Z}_j^k) \tag{22}$$

where

$$\bar{Z}_j^k = \frac{1}{k - j + 1} \sum_{i=j}^{k} Z_i(\theta_0) \tag{23}$$

and

$$Z_i(\theta_0) = \left. \frac{\partial \ln p_\theta(Y_i)}{\partial \theta} \right|_{\theta = \theta_0}. \tag{24}$$

Therefore, the GLR algorithm in (18) that provides the solution to the change detection problem in (17), can now be written as follows:

$$\begin{aligned} t_a &= \min\{k \geq 1 : g_k \geq h\} \\ g_k &\approx \max_{1 \leq j \leq k} \left\{ \frac{k - j + 1}{2} (\bar{Z}_j^k)^T I^{-1}(\theta_0) (\bar{Z}_j^k) \right\} \end{aligned} \tag{25}$$

with $\bar{Z}_j^k$ as in (23).

Under the Gaussian assumption for the vectors $\{Y_k\}$:

$$p(Y_k) = \frac{1}{\sqrt{(2\pi)^r (\det \Sigma)}} e^{-(1/2)(Y_k - \mu)^T \Sigma^{-1} (Y_k - \mu)}. \tag{26}$$

In this case, et us consider the parameter monitored for change to be the covariance Σ. The parameter Θ which is now a matrix can be treated more conveniently as a vector $\theta = col(\Theta)$, obtained by lexicographically ordering the columns of Θ, i.e., stacking the columns on top of each other. Equations (25) are used to compute the test statistic g_k at time k, and detect changes in the parameter. As in Section IV-A, let us review the data that are needed by the GLR algorithm in (25). The parameter θ_0 (covariance Σ_0) before the change must be estimated. The Fisher information matrix $I(\theta_0)$ is computed (as the covariance of the vector efficient scores, estimated at $\theta = \theta_0$). The efficient scores are computed using (24) and (26)

Let us now consider the detection of multiple scene changes in the video sequence. Similarly to the algorithm presented in Section IV-A, we will use a number M of pictures in the beginning of a video shot in order to estimate the quantities assumed known by the algorithm. The first M I- or P-pictures $\{X_k\}$ of a video shot are used for estimating the lower-dimensional subspace that enables the transformation of the original DC vectors X_k into feature vectors Y_k. As shown earlier, the parameter Θ that is monitored for change detection consists of the covariance matrix Σ corresponding to the feature vectors $\{Y_k\}$. The parameter Θ_0 before the change can be estimated using the same set M of feature vectors $\{Y_k\}$ in the beginning of a video shot, that was used for the subspace determination. This allows the computation of the vectors of efficient scores $Z_i(\theta_0)$, and their covariance $I(\theta_0)$ needed by the change detection algorithm in (25). The same block diagram and pseudocode presented in Section IV-A are relevant in this case, however, the test statistic g_k is computed differently in the nonadditive context (25).

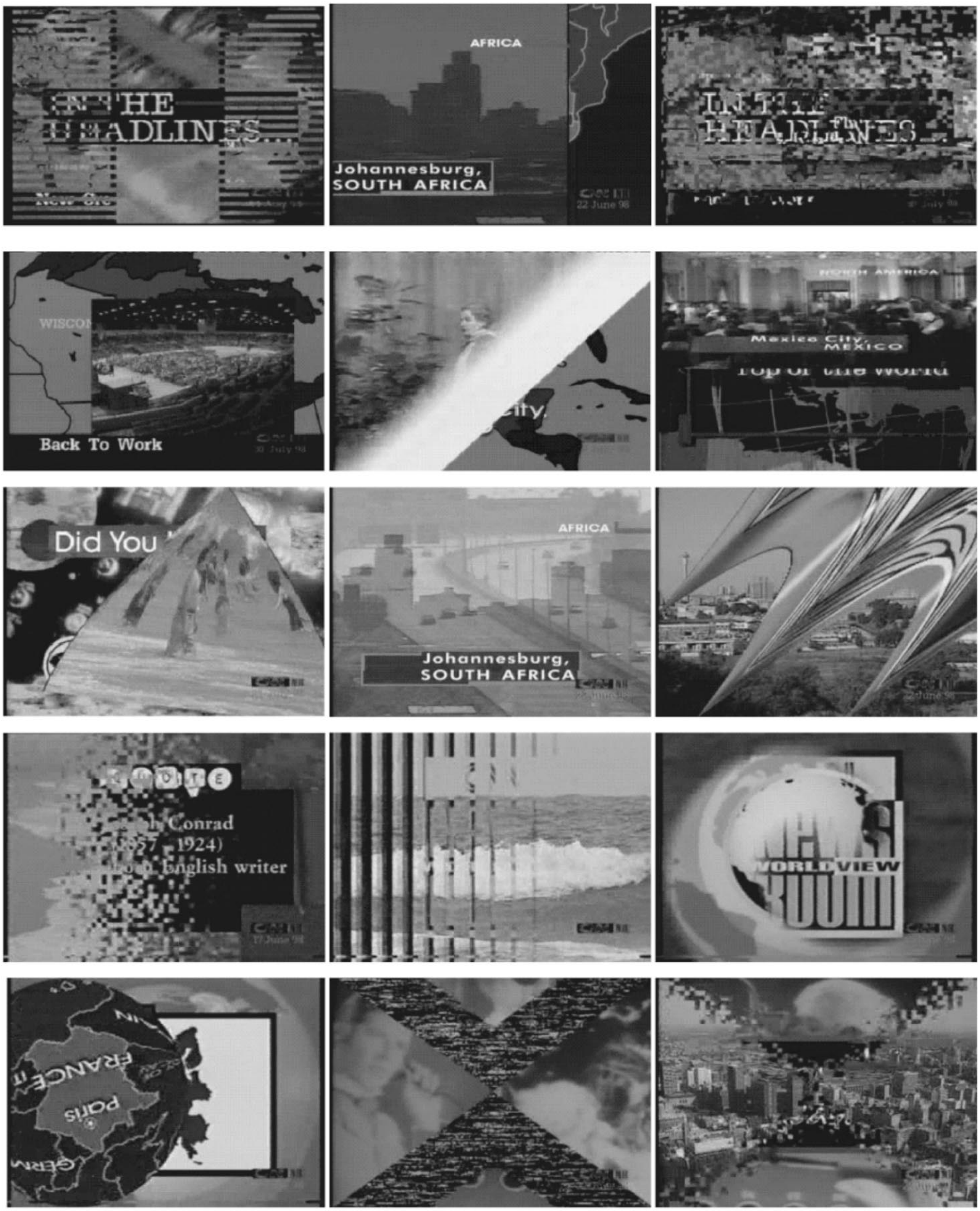


Fig. 2. Snapshots of special effect scene changes.

V. Simulation Results

The video sequences utilized in the simulations were encoded using the MPEG-2 video compression standard. The size of the video image was 240 × 352 pixels. The chrominance format of the videos was the 4:2:0 format. The encoding pattern in terms of picture types had 2 B-pictures between P-pictures.

Along with our algorithms for scene change detection, an algorithm based on the idea of color histogram differences between pictures (see Section II), and operating in the compressed domain, was also implemented. Equation (27) used for computing histogram-based measures were taken from [12], however they were used differently by applying them sequentially only to I- and P-pictures (across larger time intervals):

$$\mathrm{Sum}_c = \sum_{i=1}^{B} |H_c(k,i) - H_c(k-1,i)|$$
$$\mathrm{NormSum}_k = \frac{(\mathrm{Sum}_Y + \mathrm{Sum}_U + \mathrm{Sum}_V)^2}{BinSize} \tag{27}$$

The index c in the first equation represents one of the color components (Y,U,V), $H_c(k,i)$ represents the i^{th} histogram bin for color component c in frame k, B is the number of bins, $BinSize$ is the size of a bin, and k indexes the I- and P-pictures in the video sequence. Thus, at each I- or P-picture, absolute differences between I- or P-pictures histograms bins are summed and normalized by the histogram bin size, for the Y,U,V components. This normalized sum, $\mathrm{NormSum}_k$, is divided by the normalized sum obtained for the previous I- or P-picture to get a *ratio* value. This ratio is sequentially thresholded and if it exceeds a specified limit, a scene change is declared.

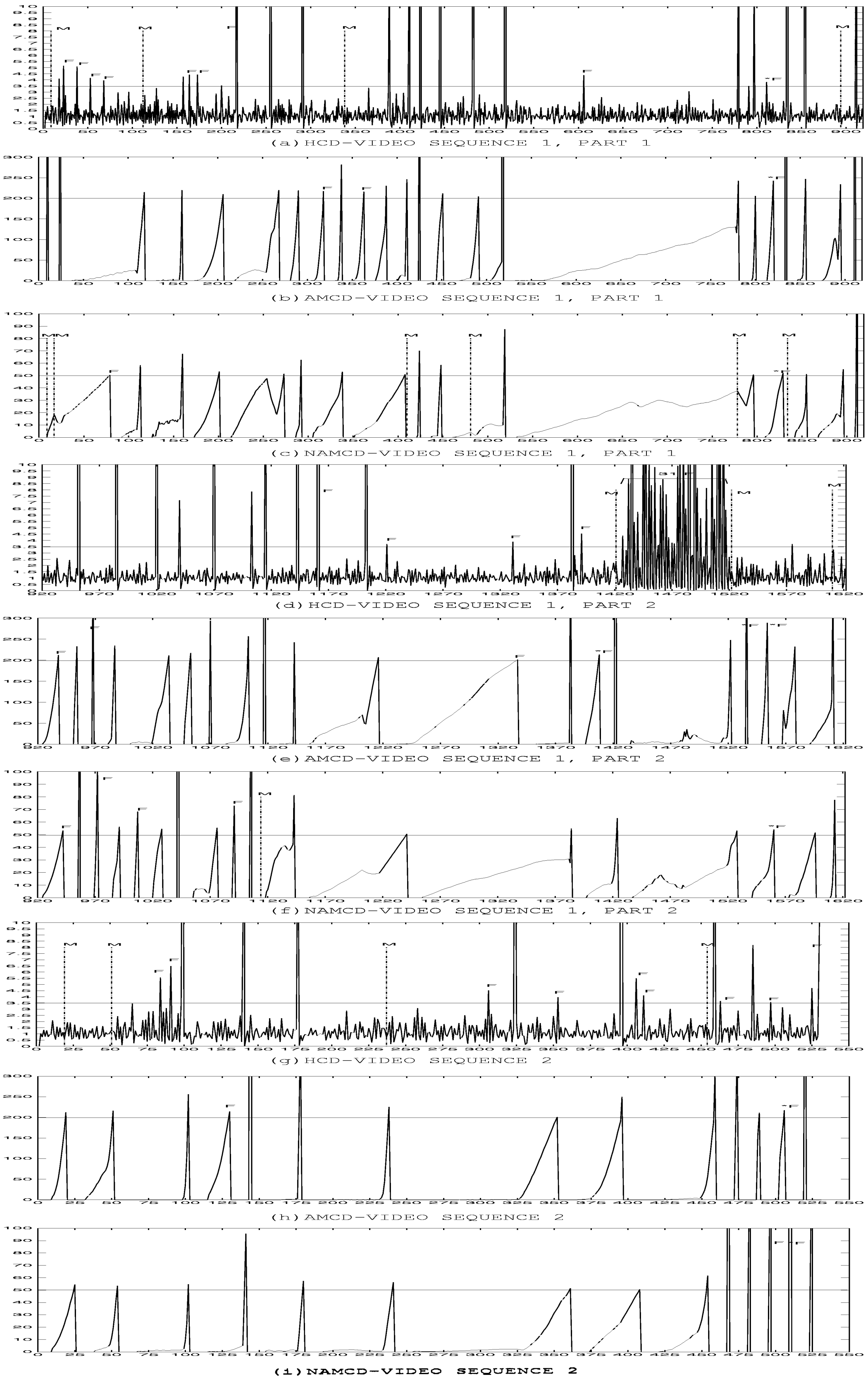
The following abbreviations will be used for the three scene change detection algorithms—HCD for the histogram-based change detector described above, AMCD for the additive model change detector, and NAMCD for the nonadditive model change detector. The algorithms presented in this paper were tested on video sequences from news (rapid succession of visual stories on various topics, no news anchor), music videos, sports, travel, documentaries (large variety of topics in a fast-paced manner of presentation). These are the types of videos where the use of gradual scene changes (special effects) is widespread, compared to movie sequences where the abrupt changes are predominant. The videos contain a large variety of changes, and a variable pattern of occurrence of these changes. Scene changes occur in fast succession in the majority of portions of the videos. Also, abrupt changes are intermixed with gradual scene changes. There is a significant degree of camera motion throughout the sequences. The video data was chosen to provide a large concentration of gradual scene changes in natural video. Thus, of the total number of scene changes, 45% are gradual. These gradual scene changes include a large variety of types. Special edits detected, samples of which are shown in Fig. 2, include vertical or horizontal band slides, coarse and additive dissolves, white-outs, fades, merging images, diagonal “wipe-out” bar, “swing-in” images, “flip page” effects, warped images, sliding, or rotating images. The common feature of these changes is that they occur gradually and thus cannot be reliably detected by examining only a few images at the time, both in terms of image data and motion vector information. Also, the lengths of the gradual changes vary widely from four to 120 frames; however, most are concentrated around 30–50 frames. The diversity of change types reflects the real case where scene changes do not obey a specific model (e.g., a linear variation in the intensity). All in all, the videos represent a difficult test set for scene change detection (see also sample execution traces provided).

Let us present the relevant parameters that were used in the implementation of the three algorithms. For the HCD, a number $B = 51$ bins (for a range 0–255) was used for each of the luminance and chrominance components. A threshold $h = 3.5$ was used. In the case of the AMCD and NAMCD algorithms we have the following settings. The number M of I- or P-pictures after each detected scene change, that are used for dimensionality reduction using the PCA is $M = 8$. In general, eigenvalues (and corresponding eigenvectors) that are smaller than 1–10% of the largest eigenvalue can be discarded (thus further reducing the number of components in the feature vector below M in some cases). After the detection of a change, a number D of pictures can be discarded in order to allow stabilization of the new video shot, before acquiring pictures for the new subspace determination. A number of $D = 5$ I- and P-pictures was used for this purpose. The maximum number L of past data points (and implicitly past I- and P-images) considered for the test statistic computation at time k (see Section IV) for both algorithms, was chosen to be $L = 300$. The threshold used for the AMCD algorithm was $h = 200$. The value of the threshold for the NAMCD algorithm was $h = 50$. The values of the thresholds in all cases were chosen based on the assessment of the performance of the algorithms on different video sequences, and to achieve a balance between false and missed alarms.

Sample execution traces of the histogram based change detection algorithm (HCD) and the traces of the test statistic g_k for our algorithms (AMCD and NAMCD), are shown in Fig. 3. The I- and P-pictures are indexed by the index k on the horizontal axes. The corresponding thresholds h are marked in each instance by a horizontal line. The traces for each of the three algorithms (HCD, AMCD, NAMCD) are presented together for each of the videos. In terms of marking alarms, all the peaks of the execution traces that exceed the thresholds and are not marked otherwise represent correct scene change detections that were ascertained by visual examination of the video sequence. False alarms are marked with the symbol “F” in the graphs. Special types of false alarms (motion-related) are marked by the symbol “$*F$.” Missed alarms are indicated by vertical lines placed at the corresponding time index position k and marked with an “M.” The results of the scene change detection are summarized in Table I.[1] The numbers in parentheses represent the performance of the algorithms if a global motion detector would invalidate alarms that are clearly created by simple camera motion (pans, zooms). HCD is not sensitive to camera motion unless it is very abrupt, as further discussed.

The HCD algorithm is able to detect abrupt changes (cuts) and some special effects that span a small number of pictures. The HCD-detected special effects feature a sufficiently abrupt transition in the grayscale/color level from one video shot to the other. The threshold for the HCD algorithm was chosen to allow it to detect the abrupt changes (with various response magnitudes) and as many as possible of the special effects, without increasing unacceptably the false alarm rate. Decreasing the value of the threshold further would drastically increase the number of false alarms. This can be easily ascertained by observing

[1] The Recall and Precision are related to two other measures used in Detection theory, i.e., Detection Rate = Recall, and False Alarm Rate = 1—Precision.



(a) HCD–VIDEO SEQUENCE 1, PART 1

(b) AMCD–VIDEO SEQUENCE 1, PART 1

(c) NAMCD–VIDEO SEQUENCE 1, PART 1

(d) HCD–VIDEO SEQUENCE 1, PART 2

(e) AMCD–VIDEO SEQUENCE 1, PART 2

(f) NAMCD–VIDEO SEQUENCE 1, PART 2

(g) HCD–VIDEO SEQUENCE 2

(h) AMCD–VIDEO SEQUENCE 2

(i) NAMCD–VIDEO SEQUENCE 2

Fig. 3. Sample execution traces for scene change detection algorithms (HCD, AMCD, NAMCD).

the "noisy" low-value portion of the HCD trace for all videos. Increasing the value of the threshold would decrease the number of gradual changes detected even further, while still not eliminating many false alarms that are represented by strong responses (see, for example, the portion $k = 1426$–1516 in Fig. 3(d)). The false alarms generated by the HCD algorithm and the ones generated by the AMCD and NAMCD algorithms are caused by different factors. The HCD is sensitive to rapid transitional changes in the image, for example changes in the color structure of the background, or flash photography. An illustrative example can be seen at $k = 1426$–1516 in Fig. 3(d), where a lot of high magnitude false alarms are generated. Once they disappear, these transitional effects leave the scene unchanged, and therefore they should not be registered as valid alarms. In the same region [$k = 1426$–1516 in Fig. 3(e) and (f)], the AMCD and NAMCD algorithms exhibit a form of statistical smoothing, by not reacting to changes that do not permanently affect the video shot—a desirable behavior. The HCD algorithm is not sensitive to gradual camera motion and/or objects gradually entering/leaving the scene. The AMCD and NAMCD algorithms react to camera motion if it causes a sufficiently large change in the image. The alarms that fall into this category are marked with the symbol "$^*F$" on the traces in Fig. 3. Depending on which course of action is taken in terms of qualifying these special type of alarms, a global motion detection module can eliminate most of these alarms, or they can be counted as valid alarms (resulting in finer-granularity segmentation). To account for a worst-case scenario, these alarms (symbol "$^*F$") were counted as *false alarms* in assessing the performance for our algorithms. Thus, the results of testing the AMCD, NAMCD algorithms represent a lower bound on their performance, and they can greatly benefit from a global motion detection. The missed alarms for the AMCD and NAMCD algorithms can be generated in some cases by an insufficient magnitude of change in the monitored parameter, or by a scene change that occurs early following another detection, while the algorithms are in the reinitialization phase.

In the simulations conducted, a value of $M = 8 \ldots 10$ I- or P-pictures was used. Considering two B pictures in between I- or P-pictures, the time interval in the beginning of each video shot used for algorithm initialization (training) is approximately 1 s at 30 fps. If another scene change occurs early during this time interval after the current detection, i.e., the current video shot is less than 1 sec. long, there will likely be a missed detection. An example of this type of situation is shown at $k = 1114(\mathrm{NAMCD})$ in Fig. 3(f). This pattern of occurrence of scene changes is more characteristic to video commercials. We would expect to have very few situations where video shots are shorter than 1 s in natural video (commercials obviously may be an exception). If one expects to use such sequences, the algorithms can be modified to use the B-pictures in the process, i.e., by taking $M = 10$ pictures in the beginning of the video shot regardless of type (I, P, B).

The algorithms presented in this paper operate sequentially and in one pass. For the AMCD and NAMCD algorithms, depending on the nature and severity of the change for a special edit, the change can be detected with zero delay, or with a delay of a number of pictures. This feature is a direct consequence of the specific sequential statistical context for change detection and it reflects the unified approach for detecting both abrupt and gradual scene changes. The statistical information about the change is allowed to accumulate and trigger a detection for scene changes that effect gradual modifications in the image.

TABLE I
CHANGE DETECTION PERFORMANCE FOR HCD, AMCD, NAMCD

	All Scene Changes (206)				Special Effects (93)	
Algorithm	# M	# F	Recall [%]	Precision [%]	# M	Recall [%]
HCD	55	111	**73**	**48**	48	**48**
AMCD	17	57 (29*)	**92**	**72** (**80***)	7	**92**
NAMCD	40	48 (30*)	**81**	**65** (**70***)	15	**84**

The differences between the execution times of the HCD, AMCD, NAMCD algorithms and the execution time of the MPEG decoder alone, were also evaluated. The dimensionality reduction presented needs to perform a PCA only for the first M I- or P-pictures in the beginning of the video shot, in the initialization phase. Thus, the new representation space is determined. Thereafter, each arriving new image in the video shot is only projected onto the determined M principal axes of the eigenspace. The GLR algorithm presented is fast, working with reduced-dimensionality data and admitting an iterative implementation whereby, at time k, only an update based on results at time $k - 1$ needs to be made. For simplicity, all execution times were computed with the algorithms in display mode, allowing full decoding and display of the video data, which evidently is not necessary for the three scene change detection algorithms (needing only minimal decoding of data). The execution time overhead added by the AMCD and NAMCD ranges from 5% to 7% of the execution time of the MPEG decoder, while the overhead added by the HCD is 12%. The additional processing time of the AMCD, NAMCD does not significantly affect the normal execution speed of the MPEG decoder.

VI. CONCLUSION

In this paper, we introduced a new real-time approach to video scene change detection using the statistical sequential analysis theory, operating on minimally-decoded video bitstreams. The algorithms presented offer an unified approach for the detection of both abrupt and gradual scene changes in natural video sequences.

For increased efficiency, the dimensionality of the original video data is reduced using an optimal transformation, that retains the most representative features of the original data, resulting in a new low-dimensional representation. The change detection algorithms operate on this transformed low-dimensional data. Scene changes are modeled as changes in parameters of a random process describing the video sequence. For video segmentation, additive and nonadditive models of scene changes are used in the context of statistical sequential analysis. No *a priori* assumptions are made about the duration of scene changes, or about specific models of the change (e.g., linear). These assumptions would limit the scope of a detection algorithm given the large variety of gradual scene changes that can be generated. In our approach, the statistical characteristics of the video data are allowed to trigger a change detection. The detection of variable-length scene changes is enabled, without the use of predetermined time windows. Through use of sufficient statistics and recursive computation of the algorithms, the limitations imposed by the need to store past frames in the video shot are eliminated.

The detection algorithms presented in this paper were tested on video sequences encoded with the MPEG-2 video compression standard. Extensions for operation with different block-based transform coding methods are straightforward; e.g., other members of the MPEG family, as well as the video-conferencing standards H.261, H.263, etc.

ACKNOWLEDGMENT

The authors would like to thank Dr. V. Poor, Department of Electrical Engineering, Princeton University, who pointed us toward the extensive literature in the area of statistical change detection theory, which forms the mathematical foundation on which our approach to video segmentation is based.

REFERENCES

[1] D. Schonfeld and D. Lelescu, "VORTEX: Video retrieval and tracking from compressed multimedia databases-multiple object tracking from MPEG-2 bitstream," *J. Vis. Commun. Image Represent.*, vol. 11, no. 2, pp. 154–182, 2000.

[2] H. J. W. Zhang, A. Kankanhalli, and S. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, 1993.

[3] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *Proc. ACM Multimedia*, San Francisco, CA, 1995, pp. 189–200.

[4] S. M.-H. Song, T.-H. Kwon, and W. M. Kim, "Detection of gradual scene changes for parsing of video data," in *Proc. Storage and Retreival for Still Image and Video Databases VI*, vol. 3312, 1998, pp. 404–413.

[5] R. Lienhart and W. Effelsberg, "On the detection and recognition of television commercials," in *Proc. Int. Conf. Multimedia Computing and Systems*, Ottawa, ON, Canada, 1997, pp. 509–516.

[6] R. Lienhart, "Comparison of automatic shot boundary detection algorithms," in *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases VII*, San Jose, CA, 1999, pp. 290–301.

[7] U. R. Gargi and S. Antani, "Performance characterization and comparison of video indexing algorithms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998, pp. 559–565.

[8] J. S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques," in *Proc. SPIE, Storage and Retrieval Image and Video Databases IV*, 1996, pp. 170–176.

[9] B. Shen, L. Dongge, and I. K. Sethi, "Cut detection via edge extraction in compressed video," in *Visual'97*, San Diego, CA, 1997, pp. 149–156.

[10] T.-S. Chua, M. Kankanhalli, and Y. Lin, "A general framework for video segmentation based on temporal multi-resolution analysis," in *Proc. Int. Workshop on Advanced Image Technology (IWAIT'2000)*, Fujisawa, Japan, 2000, pp. 119–124.

[11] V. Kobla, D. Doermann, and K. Lin, "Archiving, indexing, and retrieval of video in the compressed domain," in *Proc. SPIE Conf. Multimedia Storage and Archiving Systems*, 1996, pp. 78–89.

[12] J. Meng, Y. Juan, and S. Chang, "Scene change detection in a MPEG compressed video sequence," in *Proc. SPIE Conf. Digital Video Compression: Algorithms and Technologies*, San Jose, CA, 1995, pp. 14–25.

[13] B. Yeo and B. Liu, "Unified approach to temporal segmentation of motion JPEG and MPEG video," in *Proc. Int. Conf. Multimedia Computing and Systems*, 1995, pp. 2–13.

[14] V. Kobla, D. Doermann, and C. Faloutsos, "VideoTrails: Representing and visualizing structure in video sequences," in *Proc. ACM Multimedia Conf.*, 1997, pp. 335–346.

[15] K. Shen and J. Delp, "A fast algorithm for video parsing using MPEG compressed sequences," in *Proc. IEEE Conf. Image Processing*, 1995, pp. 252–255.

[16] W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull, "Video segmentation and classification for content based storage and retrieval using motion vectors," in *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases*, vol. 3656, San Jose, CA, 1999, pp. 687–698.

[17] R. M. Ford, C. Robson, D. Temple, and M. Gerlach, "Metrics for scene change detection in digital video sequences," *ACM Multimedia Syst.*, 2000.

[18] N. Patel and I. K. Sethi, "Compressed video processing for cut detection," *Proc. IEEE Vision, Image and Signal Processing*, vol. 143, no. 3, pp. 315–323, 1996.

[19] ——, "Video shot detection and characterization for video databases," *Pattern Recognit.*, vol. 30, no. 3, pp. 583–592, 1997.

[20] V. Kobla, D. DeMenthon, and D. Doermann, "Special effect edit detection using videotrails: A Comparison with existing techniques," in *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases VII*, San Jose, CA, 1999, pp. 302–313.

[21] B. Yeo and B. Liu, "On the extraction of DC sequence from MPEG compressed video," in *Proc. IEEE Conf. Image Processing*, vol. 2, 1995, pp. 260–263.

[22] H. Murakami and V. Kumar, "Efficient calculation of primary images from a set of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, no. 5, pp. 511–515, 1982.

[23] H. V. Poor, "Exponential quickest detection," *Annals Statist.*, vol. 26, pp. 2179–2205, 1998.

[24] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes-Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

Dan Lelescu was born in Lugoj, Romania, on June 3, 1967. He received the M.S. degree in electrical engineering from the Technical University "Politehnica" Timisoara, Romania, in 1991, and the Ph.D. degree in electrical engineering and computer science from the University of Illinois at Chicago in May 2001.

In December 2000, he joined Compression Science, Inc., Campbell, CA, where he was a Senior Scientist with the Video Group. Since January 2002, he has been a researcher with DoCoMo Communications Laboratories, San Jose CA. His research interests are in the areas of multimedia signal processing, content-based multimedia analysis and retrieval, video compression and communications, and image analysis and computer vision. He is the author of six patent applications in the areas of multimedia signal processing and video communications.

Dan Schonfeld was born in Westchester, PA, in June 1964. He received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, and the M.S. and Ph.D. degrees in electrical and computer engineering from The Johns Hopkins University, Baltimore, MD, in 1986, 1988, and 1990, respectively.

In August 1990, he joined the Department of Electrical Engineering and Computer Science, at the University of Illinois at Chicago, where he is currently an Associate Professor in the Departments of Electrical and Computer Engineering, Computer Science, and Bioengineering, and Co-Director of the Multimedia Communications Laboratory (MCL) and member of the Signal and Image Research Laboratory (SIRL).

He has authored over 60 technical papers in various journals and conferences. He has served as consultant and technical standards committee member in the areas of multimedia compression, storage, retrieval, communications, and networks. His current research interests are in multimedia communication networks; multimedia compression, storage, and retrieval; signal, image, and video processing; image analysis and computer vision; pattern recognition and medical imaging.

Dr. Schonfeld has served as an associate editor for nonlinear filtering of the IEEE TRANSACTIONS ON IMAGE PROCESSING as well as an associate editor of multidimensional signal processing and multimedia signal processing for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He was a member of the organizing committees of the IEEE International Conference on Image Processing and the IEEE Workshop on Nonlinear Signal and Image Processing. He was the plenary speaker at the INPT/ASME International Conference on Communications, Signals, and Systems.