Video Partitioning by Temporal Slice Coherency

Chong-Wah Ngo, Ting-Chuen Pong, and Roland T. Chin

***Abstract*—We present a novel approach for video partitioning by detecting three essential types of camera breaks, namely cuts, wipes, and dissolves. The approach is based on the analysis of temporal slices which are extracted from the video by slicing through the sequence of video frames and collecting temporal signatures. Each of these slices contains both spatial and temporal information from which coherent regions are indicative of uninterrupted video partitions separated by camera breaks. Properties could further be extracted from the slice for both the detection and classification of camera breaks. For example, cut and wipes are detected by color-texture properties, while dissolves are detected by statistical characteristics. The approach has been tested by extensive experiments.**

***Index Terms*—Cut detection, color-texture segmentation, dissolve detection, spatio-temporal pattern, spatio-temporal slice, video partitioning, wipe detection.**

I. Introduction

A VIDEO can be partitioned into shots; a shot is an uninterrupted segment of video frame sequence of time, space, and graphical configurations [5]. The boundary between two shots is called a camera break (or video edit). Due to the advance of video production technology, various type of video edits can be easily created to indicate the change of space and time, or to highlight important events. For instance, sport videos often use a special-effect edit between the live footage and instant-replay to intensify impression. Therefore, by detecting, as well as classifying, camera breaks, we can facilitate the content analysis, indexing, and browsing of video data, and in addition, reduce video retrieval problems to image (or key-frame) retrieval problems.

Based on the transitional properties of video edits, there are three major types of camera breaks: *cut*, *wipe*, and *dissolve*. A camera cut is an instantaneous change from one shot to another; a wipe is a moving transition of a frame (or a pattern) across the screen that enables one shot to gradually replace another; and a dissolve superimposes two shots where one shot gradually appears while the other fades out slowly. Fig. 1 shows examples of the three types of camera breaks. Since frames located at the boundaries of wipe or dissolve can not represent the content of a shot, in principle, it is necessary to separate those frames from shots.

In the current literature, there are various algorithms for detecting camera breaks. Work on camera-cut detection include [9]–[11], [13], [18], [19], [21], wipe detection include [2], [12], [16], and dissolve detection include [3], [10], [12], [19], [21]. Wipes and dissolves involve gradual transitions with no drastic changes between two consecutive frames, and hence, are relatively difficult to identify. While cuts can be identified by comparing two adjacent frames, wipes and dissolves require the investigation of frames along a larger temporal scale.

In general, most cut detection algorithms can segment a video into shots accurately if the sequence has smooth within-shot frame transitions and abrupt between-shot spatial changes. The speed efficiency of these algorithms are normally improved by either processing in the compressed domain (e.g., MPEG) [10], [11], [19], [21] or sub-sampling of the spatial and temporal of video frames [18]. Features extracted from the compressed domain are rich in both global and local properties and are ideal for cut detection. On the contrary, the video sub-sampling scheme depends on the spatial window size and the temporal sub-sampling rate. and has shown to be sensitive to object and camera motions.

Although there exists many cut detection algorithms, there are relatively few wipe- and dissolve detection algorithms proposed in the literature. Wu *et al.* [16] proposed the projected pairwise difference deviation to detect wipes. However, it can only handle very limited, yet simple, wipe patterns. Alattar [2] proposed a more general wipe detection algorithm by analyzing the statistical change in mean and variance of the wiped frames. However, this statistical approach assumes that there is only slight motion in shots, so that the beginning and ending of wipe regions can be identified. As a result, it can not detect wipes in videos with fast motions. Meng *et al.* [10] and Alattar [3] proposed dissolve detection algorithms by looking for the parabolic functions of intensity variance in the dissolve regions. These algorithms assume that dissolves are linear, and hence, can only tolerate slight motions during the dissolve periods.

In this paper, we proposed to detect camera cuts, wipes, and dissolves based on a spatio-temporal slice model. The model is built by constructing a spatio-temporal slice of the video and analyzing its temporal coherency. Slice coherency is defined as the logical consistency of an event in a shot which is referred to as the common rhythm shared by all frames within a shot. A camera break is detected if there is a change of rhythm. Early work in the temporal slice analysis is mainly on motion estimation [1], [14], [20], while our approach focuses on developing algorithms to measure the change of rhythms for the cut, wipe, and dissolve detections. Compared to other camera-break detection algorithms, our proposed algorithm handles fast motions and color changes within a shot. In addition, it is capable of detecting various wipe patterns. The proposed dissolve detection method is similar to [10], [3], except that the statistical features are computed directly from the temporal slices.

Manuscript received October 18, 1999; revised May 11, 2001. This work was supported in part by RGC Grant HKUST661/95E, Grant HKUST6072/97E, and Grant HKUST6089/99E. This paper was recommended by Associate Editor H. J. Zhang.

The authors are with the Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: cwngo@cs.ust.hk; tcpong@cs.ust.hk; roland@cs.ust.hk).

Publisher Item Identifier S 1051-8215(01)06531-4.

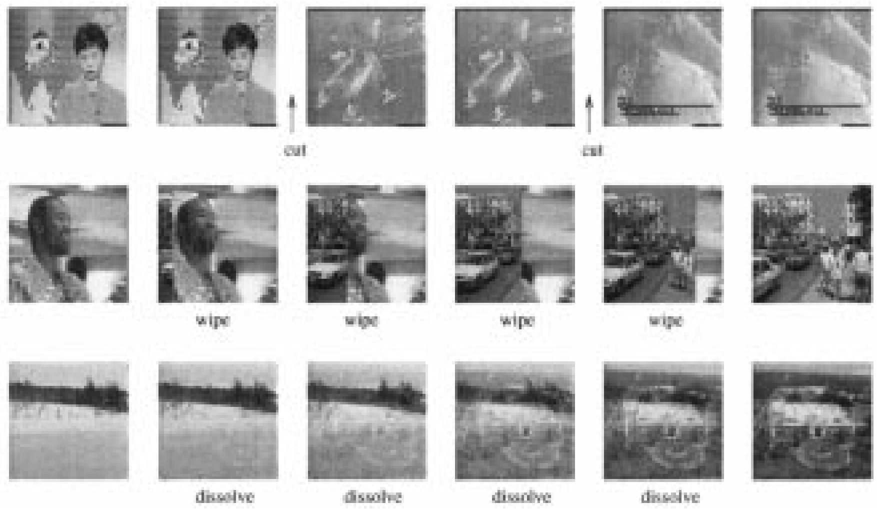


Fig. 1. Three types of camera breaks.

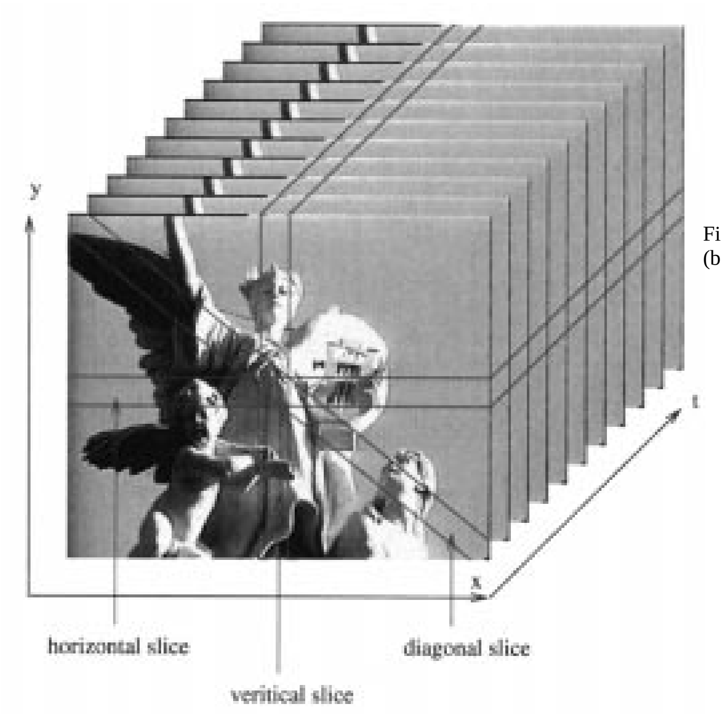


Fig. 2. Three spatio-temporal video slices taken from an image volume along the temporal dimension.

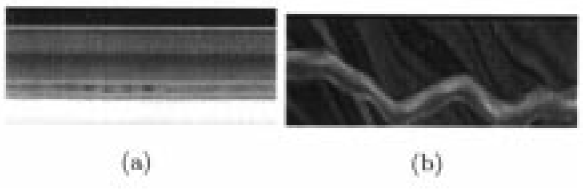
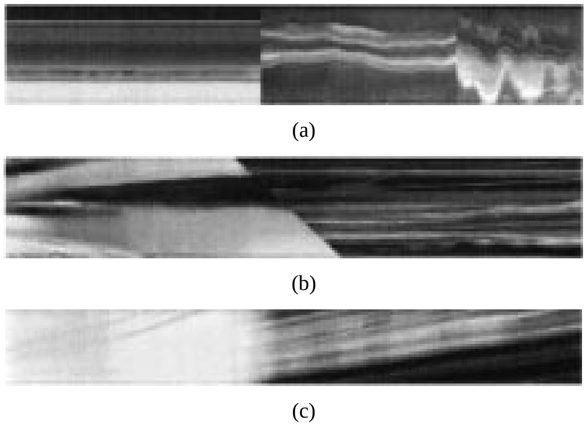


Fig. 3. Spatio-temporal slices of different rhythms. (a) Stationary scene. (b) Fast motion.



(a)

(b)

(c)

Fig. 4. Samples of spatio-temporal slices. (a) Three shots connected by two cuts. (b) Two shots connected by a wipe. (c) Two shots connected by a dissolve.

A. Concept of Temporal Slice Coherency

Fig. 2 shows a video sequence arranged as a volume with (x, y) representing image dimensions and t temporal dimension. We can also view the volume as formed by a set of spatio-temporal 2-D slices, each with dimension (x, t) or (y, t), for example. Each spatio-temporal slice is then a collection of scans[1] in the same selected position of every frame as a function of time. The spatio-temporal slice is used to extract an indicator to capture the coherency of the video. In Fig. 3, we show two spatio-temporal slices of different coherent rhythms. A shot without motion will have horizontal lines running across the spatio-temporal slice, while a shot with fast motion will create oscillatory patterns.

Fig. 4 shows three spatio-temporal slices. Each slice contains several spatially uniform color-texture regions, and each region is considered to have a unique rhythm. The boundary of regions which shows a distinct change of rhythm indicates the presence of a camera break. The shape and orientation of the boundary are affected by the types of camera breaks: a cut results in a vertical boundary line, a wipe results in a slanted boundary line, and a dissolve results in a slow transition which shows a burred boundary.

It becomes obvious now that shot boundaries can be detected and classified by segmenting a spatio-temporal slice into regions each of a uniform rhythm. Compared with other existing

[1] A scan is defined as a strip of an image. For example, it can be a row or a column in an image frame.

approaches, our proposed approach offers the following advantages.

1) Camera-break detection is reduced to image segmentation. By selecting a small subset of spatio-temporal slices for segmentation, the processing of the whole image volume is reduced to a few 2-D slices.
2) By analyzing the properties of regional boundaries, we can detect, as well as classify, various types of camera breaks.
3) By locating the two end points of a regional boundary, we can detect the start and end time of a wipe.
4) Compared with the wipe detection algorithms proposed in [2] and [16], our approach does not require performing global motion compensation explicitly in order to distinguish between wipe and camera motion.

In this paper, we propose two different measures to detect the breaks between coherent shots in a video life. First, the change of rhythm of a spatio-temporal slice due to a cut or a wipe is measured by the change of color-texture properties of the slice. Second, the change of rhythm due to a dissolve is measured by the statistical changes in the temporal slices. Since the rhythm of the two adjacent shots are intertwined during a dissolve where the change in coherency cannot be easily distinguished by color-texture properties.

This paper is organized as follows. Section II presents methods on the efficient computation of the three types of spatio-temporal slices. Section III proposes a spatio-temporal slice model which captures the shape of regional boundary as *a priori* knowledge and segments the spatio-temporal slices into regions. Sections IV and V discuss the cut and wipe detection algorithms based on the proposed spatio-temporal slice model. Section VI describes a statistical approach for detecting dissolves. Section VII presents experimental results, while Section VIII discusses the pros and cons of various tested algorithms. Section VIII concludes the paper.

II. Computation and Patterns of Spatio-Temporal Slices

The size of a video sequence (the volume as in Fig. 2) is reduced by replacing each full size image with dc image.[2] The dc sequence is obtained directly from an MPEG video without decompression[3]. This offers two advantages: computational efficiency, since the volume is reduced by 64 times, and the image volume is inherently smoothed.

Let T be the number of images in a volume and f_{dc} be a dc image of size $M \times N$. Our approach projects the 2-D image f_{dc} vertically, horizontally and diagonally to three 1-D scans. The value of a pixel i in a scan is computed by

$$h_i = \sum_{p=k_2-j}^{k_2+j} \alpha_p f_{\mathrm{dc}}(i,p), \quad \text{where } k_2 = \frac{N}{2} \tag{1}$$

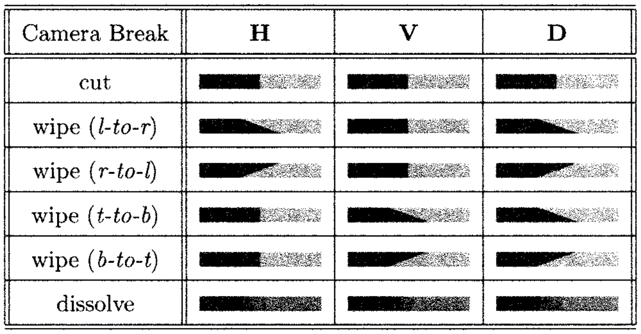


Fig. 5. Spatio-temporal slice patterns generated by various types of camera breaks: *l-to-r* (left-to-right); *r-to-l* (right-to-left); *t-to-b* (top to bottom); *b-to-t* (bottom-to-top).

$$v_i = \sum_{p=k_1-j}^{k_1+j} \alpha_p f_{\mathrm{dc}}(p,i), \quad \text{where } k_1 = \frac{M}{2} \tag{2}$$

$$d_i = \sum_{p=i-j}^{i+j} \alpha_p f_{\mathrm{dc}}(p,i,t) \tag{3}$$

where $0 \leq p < M$ or N and j is the window of support for the weighted projection of f_{dc} onto a scan line. The coefficients α_p are selected weights of the linear projection where $\sum \alpha_p = 1$. When $j = 0$, α_p is set to 1 and the projection is simply the scan which passes through the center of the image f_{dc}. To ensure smoothness of the scans, we select $j = 1$ to perform Gaussian smoothing on the dc data, where $[\alpha_p] = [0.2236, 0.5477, 0.2336]$ [4]. By cascading these scans over time, we acquire a 2-D spatio-temporal slice $\mathbf{H}$ (size $M \times T$) formed by the horizontal scans, a 2-D image $\mathbf{V}$ (size $N \times T$) formed by the vertical scans, and a 2-D image $\mathbf{D}$ (size $N \times T$) formed by diagonal scans. Fig. 2 has shown the three spatio-temporal slices of a video sequence.

There are two questions associated with this approach: 1) the number of spatio-temporal slices in a volume that should be taken for analysis and 2) the strategy on how to select these spatio-temporal slices. In principle, selecting more slices will improve the detection results at the expense of computational cost. To be most efficient, two spatio-temporal slices of orthogonal directions are necessary where $\mathbf{H}$ captures the temporal coherency in the horizontal direction and $\mathbf{V}$ models the temporal coherency in the vertical direction. For our application, in order to detect and classify different camera breaks, three slices ($\mathbf{H}, \mathbf{V}$, $\mathbf{D}$) are used to provide the necessary discriminatory power to resolve the various types of breaks. The three selected scans are chosen to be located at the center of the dc image for convenience. Fig. 4 shows a sample set of $\{\mathbf{H}, \mathbf{V}, \mathbf{D}\}$, which will serve as models for the segmentation algorithm of the spatio-temporal slices.

III. Spatio-Temporal Slice Model

To effectively segment a spatio-temporal slice into coherent regions, on one hand we need to extract features that represent coherency, while on the other hand, we need to model the change of coherency at the regional boundaries. We propose a model that extracts the color-texture features from slices and captures

[2] A dc image is formed by using the first coefficient of each 8×8 discrete cosine transform (DCT) block.

[3] The estimation of dc sequence from the P- and B-frames of a MPEG has been discussed in [17].

the shape and orientation of regional boundary (as shown in Fig. 5) as model energy. A regional boundary is detected when there is a violation of color-texture coherency in the spatio-temporal slices extracted from a video sequence.

For ease of understanding, the following mathematical notations are used in the remaining paper:

1) $\mathbf{H} = [H_r, H_g, H_b, H_y]$, $\mathbf{V} = [V_r, V_g, V_b, V_y]$ and $\mathbf{D} = [D_r, D_g, D_b, D_y]$ are denoted as the spatio-temporal slices in (r, g, b) color space[4] and y luminance space. Each H_c (or V_c or D_c) is indexed by space i and time t ($H_c(i,t)$ for instance), and $c \in \{r, g, b\}$.
2) h_t, v_t, and d_t are defined as the scans in $\mathbf{H}$, $\mathbf{V}$ and $\mathbf{D}$, respectively. Each scan is in fact one column of a slice indexed by space i at time t.
3) $h_t(i)$, which is a pixel in the scan h_t, is abbreviated to h_i (similarly for v_i and d_i).
4) $\xi \in \{\text{cut}, \text{wipe}\}$ is defined as shot boundary and f_t is denoted as an image frame at time t. We write $f_t \in \xi$ if f_t is at the boundary of two shots.
5) In spatio-temporal slices, ξ is also referred to as the boundary of two connected regions. In this case, we write $h_t \in \xi$ $(h_i \in \xi)$, for example, to indicate a scan (a pixel) is at the regional boundary.

A. Computing Color-Texture Feature

The color edge information of a spatio-temporal slice is computed by

$$E_{\sigma,\theta}^{H_c} = \bar{\mathbf{G}}'_{\sigma,\theta} * H_c \tag{4}$$

where $*$ is a convolution operator. $\bar{\mathbf{G}}'_{\sigma,\theta}$ is the first derivative Gaussian along the direction θ given by

$$\bar{\mathbf{G}}'_{\sigma,\theta}(i,t) = -\frac{x}{\sigma^2}\bar{\mathbf{G}}_{\sigma,\theta}(i,t), \quad \bar{\mathbf{G}}_{\sigma,\theta}(i,t) = \bar{\mathbf{G}}_{\sigma}(i',t') \tag{5}$$

where $= i\cos\theta + t\sin\theta$ and $= -i\sin\theta + t\cos\theta$. $\bar{\mathbf{G}}_{\sigma}(i,t) = \exp\{-(i^2 + t^2)/(2\sigma^2)\}$ is a Gaussian filter controlled by a smoothing parameter σ.

The texture feature is computed based on the Gabor decomposition [7]. The idea is to decompose images into multiple spatial-frequency channels, and to use the real components of channel envelopes to form a feature vector. The complex Gabor images are

$$T_{\sigma_i,\sigma_t,\theta} = \hat{\mathbf{G}}_{\sigma_i,\sigma_t,\theta} * H_y. \tag{6}$$

The Gabor filter $\hat{\mathbf{G}}_{\sigma_i,\sigma_t,\theta}(x,y) = \hat{\mathbf{G}}_{\sigma_i,\sigma_t}(i',t')$ is expressed as

$$\hat{\mathbf{G}}_{\sigma_i,\sigma_t}(i,t) = \left(\frac{1}{2\pi\sigma_i\sigma_t}\right)\exp\left\{-\frac{1}{2}\left(\frac{i^2}{\sigma_i^2} + \frac{t^2}{\sigma_t^2}\right)\right\} \times \exp\{2\pi jWt\} \tag{7}$$

where

$j \quad = \sqrt{-1};$
$W \quad = \sqrt{u^2 + v^2};$
(u,v) center of the desired frequency.

We empirically set $\theta = \{0°, -45°, 45°\}$, $u = v = 0.4$ and fix the values of σ, σ_i and σ_t. As a result, the color-texture feature at each pixel is a 12-D feature vector. For instance, the feature vector of a pixel at H_r is in the form $[E_{\sigma,\theta}^{H_r}(i,t), E_{\sigma,\theta}^{H_g}(i,t), E_{\sigma,\theta}^{H_b}(i,t), T_{\sigma_i,\sigma_t,\theta}(i,t)]$, where $\theta = \{0°, -45°, 45°\}$.

B. Formulating Model Energy

The probability that a frame f_t is at the boundary of two shots ξ can be written as[5]

$$\begin{aligned} p(f_t \in \xi \,|\, \mathbf{H}, \mathbf{V}, \mathbf{D}) &= p(h_t \in \xi \,|\, \mathbf{H}) p(v_t \in \xi \,|\, \mathbf{V}) p(d_t \in \xi \,|\, \mathbf{D}) \\ &= \sum_i p(h_i \in \xi \,|\, \mathbf{H}) p(v_i \in \xi \,|\, \mathbf{V}) p(d_i \in \xi \,|\, \mathbf{D}). \end{aligned} \tag{8}$$

Combined with the local characteristic of Markov random field [8], we can model the local spatio-temporal configuration of h_i, v_i, and d_i as

$$p(f_t \in \xi \,|\, \mathbf{H}, \mathbf{V}, \mathbf{D}) - \sum_i p(h_i \in \xi \,|\, \mathbf{H}_N) p(v_i \in \xi \,|\, \mathbf{V}_N) p(d_i \in \xi \,|\, \mathbf{D}_N) \tag{9}$$

where $\mathbf{H}_N$, $\mathbf{V}_N$ and $\mathbf{D}_N$ are 3×3 neighborhood systems that will be described in Section III-C. Due to the Markov–Gibbs equivalent [8], we can assume that $p(h_i \in \xi \,|\, \mathbf{H}_N)$, $p(v_i \in \xi \,|\, \mathbf{V}_N)$ and $p(d_i \in \xi \,|\, \mathbf{D}_N)$ follow Gibbs distribution. Hence, we have

$$p(h_i \in \xi \,|\, \mathbf{H}_N) = \frac{1}{Z}\exp\{-U(h_i)\} \tag{10}$$

where Z is a normalizing constant, and U is an energy function defined by the neighborhood system ($p(v_i \in \xi \,|\, \mathbf{V}_N)$ and $p(d_i \in \xi \,|\, \mathbf{D}_N)$ also have a similar formula as (10)). Substituting (10) into (9) and taking the logarithm on both sides, we have

$$\begin{aligned} \log\{p(f_t \in \xi \,|\, \mathbf{H}, \mathbf{V}, \mathbf{D})\} &\propto -\sum_i \{U(h_i) + U(v_i) + U(d_i)\} \\ L(f_t \in \xi) &\propto -\sum_i \{U(h_i) + U(v_i) + U(d_i)\} \end{aligned} \tag{11}$$

where $L(f_t \in \xi) = \log\{p(f_t \in \xi \,|\, \mathbf{H}, \mathbf{V}, \mathbf{D})\}$. In other words, the likelihood of a camera break at f_t is dependent on the total energy of the scans at time t.

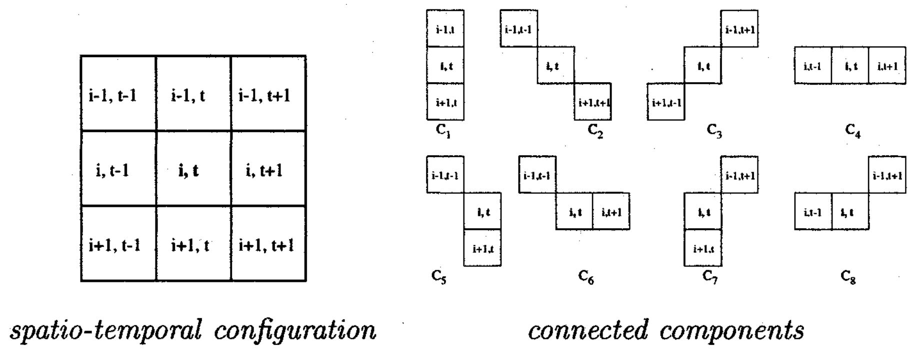
C. Segmenting Spatio-Temporal Slices

From (11), we further classify the energy function U to three types of energy: U_{cut}, U_{wipe^-}, and U_{wipe^+}, where

$$\begin{aligned} U_{\text{cut}} &= \{U_{\text{cut}}^r, U_{\text{cut}}^g, U_{\text{cut}}^b, U_{\text{cut}}^y\} \\ U_{\text{wipe}^-} &= \left\{U_{\text{wipe}^-}^r, U_{\text{wipe}^-}^g, U_{\text{wipe}^-}^b, U_{\text{wipe}^-}^y\right\} \\ U_{\text{wipe}^+} &= \left\{U_{\text{wipe}^+}^r, U_{\text{wipe}^+}^g, U_{\text{wipe}^+}^b, U_{\text{wipe}^+}^y\right\}. \end{aligned}$$

[4] Note that MPEG uses YCrCb color space. Our method converts the YCrCb to RGB components.

[5] $\mathbf{H}$, $\mathbf{V}$, and $\mathbf{D}$ are assumed independent since they are extracted from an image volume through different orientations



(a) (b)

Fig. 6. Neighborhood system of a pixel h_i (or $h_t(i)$). (a) Spatio-temporal configuration. (b) Connected components.

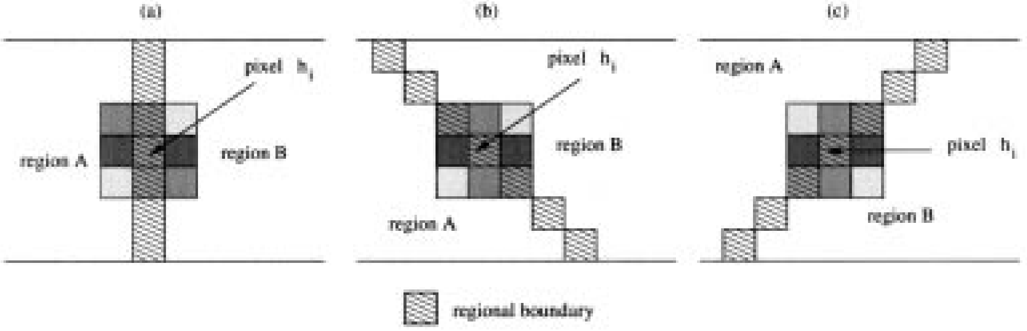


Fig. 7. An illustration for the energy computation of (12). The connected components which cross both the region A and region B will have higher energy of $\Gamma^r_{C_j}$ than the connected component which is on the regional boundary. As a result, in (a), U^r_{cut} will have relatively low energy, in (b), $U^r_{\text{wipe}-}$ will have relatively low energy, and in (c), $U^r_{\text{wipe}+}$ will have relatively low energy.

Each energy type has four elements describing their color-texture properties in (r, g, b) color space and y luminance space. U_{cut} models the energy value of a vertical boundary line and U_{wipe^+} models the energy value of a slanted boundary line of positive slope, while U_{wipe^-} models the energy value of a slanted boundary line of negative slope. For simplicity, we only describe the energy formulation of $U^r_{\text{cut}}, U^r_{\text{wipe}^+}, U^r_{\text{wipe}^-}$. The energy function of other slices are computed in a similar way.

The energy of a pixel h_i is computed based on the configuration of a neighborhood system,[6] as shown in Fig. 6. We define eight connected components $C = \{C_1, C_2, \ldots, C_8\}$ in the system to characterize h_i. Each component describes the spatio-temporal relationship of h_i with its neighboring pixels. Except for C_4, which represents a horizontal boundary, the connected components describe the shape of the regional boundaries of interest to our camera break detection.

Based on the neighborhood system, we define

$$\begin{bmatrix} U^r_{\text{cut}}(h_i) \\ U^r_{\text{wipe}^-}(h_i) \\ U^r_{\text{wipe}^+}(h_i) \end{bmatrix} = 3\begin{bmatrix} \Gamma^r_{C_1}(h_i) \\ \Gamma^r_{c'}(h_i) \\ \Gamma^r_{c''}(h_i) \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Gamma^r_{C_1}(h_i) \\ \Gamma^r_{C'}(h_i) \\ \Gamma^r_{C''}(h_i) \end{bmatrix} - \begin{bmatrix} \Gamma^r_{C_4}(h_i) \\ \Gamma^r_{C_4}(h_i) \\ \Gamma^r_{C_4}(h_i) \end{bmatrix} \quad (12)$$

where

$$\Gamma^r_{c'}(h_i) = \min_{c \in \{C_2, C_5, C_6\}} \Gamma^r_c(h_i)$$
$$\Gamma^r_{c''}(h_i) = \min_{c \in \{C_3, C_7, C_8\}} \Gamma^r_c(h_i).$$

$\{C_2, C_5, C_6\}$ are the connected components with a negative slope, while $\{C_3, C_7, C_8\}$ are the connected components with a positive slope. $\Gamma^r_{C_j}(h_i)$ is a potential energy computed by the sum of absolute feature values difference in the connected component C_j. Denote $\eta_1 = H_r(i_1, t_1)$, and $\eta_2 = H_r(i_2, t_2)$ as the neighbors of h_i such that $\{\eta_1, h_i, \eta_2\}$ forms a connected component C_j. The potential energy which represents the edge information at h_i in the color space r is

$$\Gamma^r_{C_j}(h_i) = \left|E^{H_r}_{\sigma,\theta}(i,t) - E^{H_r}_{\sigma,\theta}(i_1,t_1)\right| + \left|E^{H_r}_{\sigma,\theta}(i,t) - E^{H_r}_{\sigma,\theta}(i_2,t_2)\right| \quad (13)$$

where $\theta = 0°$ for U_{cut}, $\theta = -45°$ for U_{wipe^-}, and $\theta = 45°$ for U_{wipe^+}. When formulating the potential energy which represents the texture information in the luminance space y, (13) is modified to

$$\Gamma^y_{C_j}(h_i) = \left|T^{H_y}_{\sigma,\theta}(i,t) - T^{H_y}_{\sigma,\theta}(i_1,t_1)\right| + \left|T^{H_y}_{\sigma,\theta}(i,t) - T^{H_y}_{\sigma,\theta}(i_2,t_2)\right|. \quad (14)$$

Fig. 7 illustrates the intuitive meaning of (12). On one hand, negative weights are given to the potential energies of connected com-

[6] $\mathbf{H}_N, \mathbf{V}_N, \mathbf{D}_N$ have a same neighborhood configuration.

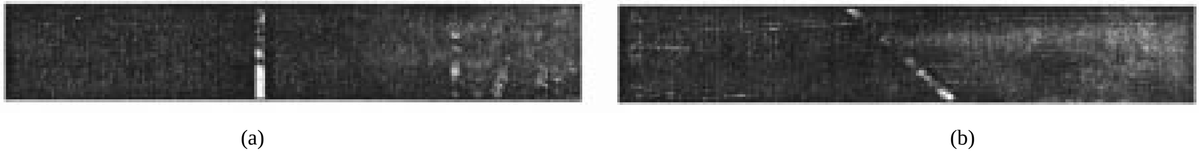


Fig. 8. Computed energy for the spatio-temporal slices shown in Fig. 4. (a) U_{cut} of horizontal slices. (b) U_{wipe} of horizontal slices.

ponents which cross two distinct regions, since they consist of relatively high values; on the other hand, positive weights are given to those components which are inside one region or along the regional boundaries. When all the potential energies are summed, $U^r_{\text{cut}}, U^r_{\text{wipe}^-}$ and $U^r_{\text{wipe}^+}$ will give low energy values when the regional boundaries of the preferred direction are encountered.

Finally, the energies computed from the color and luminance spaces are combined to form

$$U_{\text{cut}}(h_i) = \min_{j\in\{r,g,b\}} U^j_{\text{cut}}(h_i) + U^y_{\text{cut}}(h_i) \tag{15}$$

$$U_{\text{wipe}^-}(h_i) = \min_{j\in\{r,g,b\}} U^j_{\text{wipe}^-}(h_i) + U^y_{\text{wipe}^-}(h_i) \tag{16}$$

$$U_{\text{wipe}^+}(h_i) = \min_{j\in\{r,g,b\}} U^j_{\text{wipe}^+}(h_i) + U^y_{\text{wipe}^+}(h_i) \tag{17}$$

where the energy computed through color edge and texture features are equally weighted. Fig. 8 shows the segmentation results, the white lines which indicate the presence of low energy run across the boundaries of connected regions.

IV. Cut Detection

From (11), let the regional boundary $\xi = \text{cut}$ and $U = U_{\text{cut}}$, we have

$$L(f_t \in \text{cut}) \propto -\sum_i \{U_{\text{cut}}(h_i) + U_{\text{cut}}(v_i) + U_{\text{cut}}(d_i)\}. \tag{18}$$

It is obvious that cuts can be located by looking for scans possessing lower energy than a pre-defined threshold. However, such a simple scheme will normally fail because it is difficult to find a threshold that can tolerate both false and missed detections. Therefore, cuts are detected by looking for the local minimals of energy value in our implementation. The idea is adopted from Ferman and Tekalp [6], which uses the temporal filtering techniques to enhance the values of local maximals. This idea allows us to not perform shot pruning as proposed in [11].

V. Wipe Detection

Detection of wipes is more complicated than cuts due to the variety of wipe patterns (see Fig. 11). Let the regional boundary $\xi = \text{wipe}$, we write (11) as

$$L(f_t \in \text{wipe}) \propto -\min \left\{ \begin{array}{l} \sum_i \{U_{\text{wipe}}(h_i) + U_{\text{wipe}}(v_i) + U_{\text{wipe}}(d_i)\} \\ \sum_i \{U_{\text{cut}}(h_i) + U_{\text{wipe}}(v_i) + U_{\text{wipe}}(d_i)\} \\ \sum_i \{U_{\text{wipe}}(h_i) + U_{\text{cut}}(v_i) + U_{\text{wipe}}(d_i)\} \\ \sum_i \{U_{\text{wipe}}(h_i) + U_{\text{wipe}}(v_i) + U_{\text{cut}}(d_i)\} \end{array} \right\} \tag{19}$$

where

$$U_{\text{wipe}} = \sqrt{U^2_{\text{wipe}^-} + U^2_{\text{wipe}^+}}. \tag{20}$$

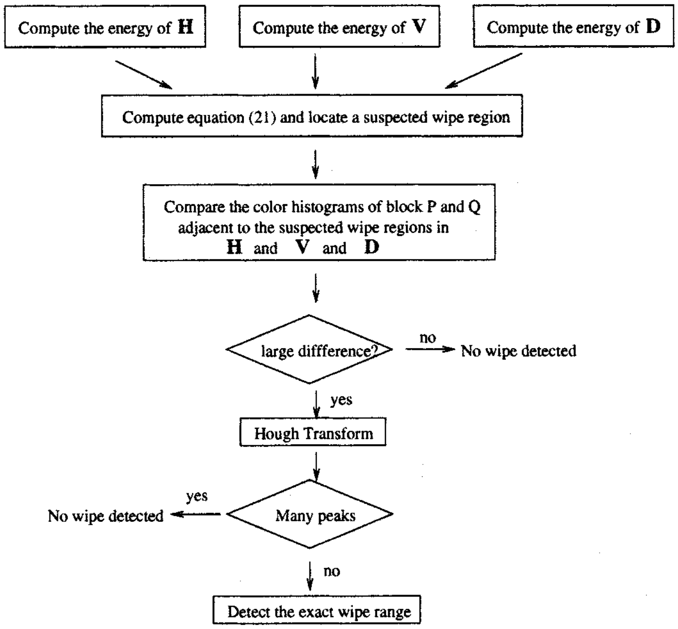

Compute the energy of H
Compute the energy of V
Compute the energy of D
Compute equation (21) and locate a suspected wipe region
Compare the color histograms of block P and Q adjacent to the suspected wipe regions in H and V and D
large diffference?
no
No wipe detected
yes
Hough Transform
Many peaks
yes
No wipe detected
no
Detect the exact wipe range

Fig. 9. Wipe detection algorithm.

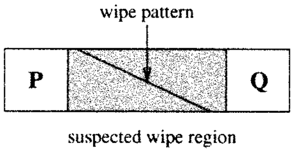


Fig. 10. P and Q are the blocks (formed by five scans) adjacent to the suspected wipe range.

The energy configuration of $L(f_t \in \text{wipe})$ can cover different wipe patterns. $L(f_t \in \text{wipe})$ will pick up the lowest energy which best fits the wipe pattern under investigation. Wipe detection, in contrast to cut detection, can not be easily achieved by investigating the energy value of every scan independently. Instead, the total energy in a group of five adjacent scans is summed when locating wipes. Fig. 9 depicts our wipe detection algorithm for a suspected wipe pattern illustrated in Fig. 10. It starts by computing the energy of three spatio-temporal slices, and then locates the suspected wipe regions. The color histograms of the two neighboring blocks (blocks P and Q, as shown in Fig. 10) of the suspected wipe regions in $\mathbf{H}, \mathbf{V}$ and $\mathbf{D}$ are compared.[7] If the histogram difference is larger than an empirical threshold, Hough transform [15] will be performed to locate the boundary lines formed by the wipe transitions. These lines correspond to the local peaks in the Hough space. Only pixels whose values exceed 0.05% of the total values in the Hough space are considered as peaks. If 10% of the total

[7] The sizes of the suspected wipe regions in $\mathbf{H}, \mathbf{V}$ and $\mathbf{D}$ are not necessary to be equal, the sizes will be adjusted so that only the regions with low energy values of $U_{\text{wipe}}(\cdot)$ will be considered.

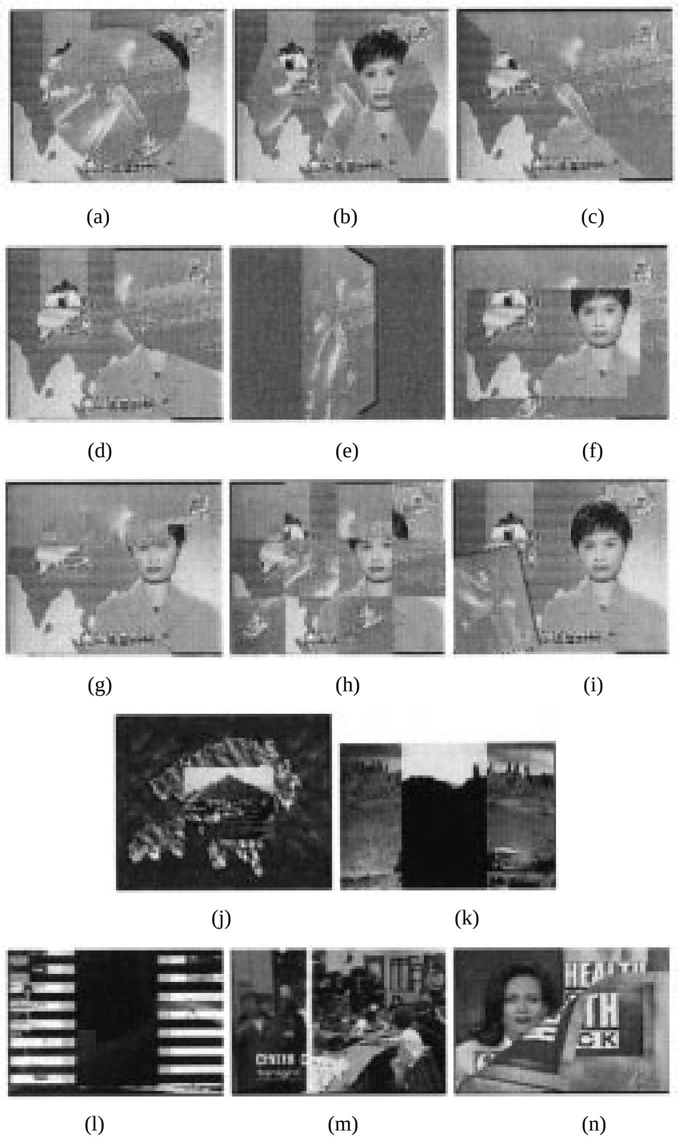

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n)

Fig. 11. Various wipe transitions. (a) Iris round. (b) Irish shape. (c) Radial wipe. (d) Clock wipe. (e) Flip over. (f) Spiral box. (g) Zig-zag blocks. (h) Checker wipe. (i) Motion wipe. (j) Zoom wipe. (k) Barn door. (l) Band wipe. (m) Wipe. (n) Page turn.

pixels are peaks,[8] the suspected wipe regions are regarded as object or camera motions.[9]

The duration of a detected wipe range is obtained directly from a peak with the highest value in the Hough space. Let the detected wipe range in $\mathbf{H}$ as $\tau^h = [t_1^h, \ldots, t_p^h]$, $\mathbf{V}$ as $\tau^v = [t_1^v, \ldots, t_q^v]$, and $\mathbf{D}$ as $\tau^d = [t_1^d, \ldots, t_r^d]$. If $\tau^h \cap \tau^v \cap \tau^d \neq \emptyset$, the start of a wipe transition is detected as $\min(t_1^h, t_1^v, t_1^d)$, while the end of a wipe transition is detected as $\max(t_p^h, t_q^v, t_r^d)$. A wipe is also detected similarly if $\tau^h \cap \tau^v \neq \emptyset$ or $\tau^h \cap \tau^d \neq \emptyset$ or $\tau^v \cap \tau^d \neq \emptyset$. In addition, two wipes are merged if they are less than 15 frames apart.

VI. Dissolve Detection

A dissolve connects the boundaries of two shots smoothly; as a result, the connected shots share a blurred boundary region in a spatio-temporal slice. Globally, the slice is composed of two regions with different visual appearances; locally, it exhibits a smooth transition from one region to another. Our dissolve detection algorithm is similar to [3], [10], except that it computes the statistical characteristic of intensity values directly on three spatio-temporal slices rather than individual frames.

Denote $\mathrm{Dissolve}(x, y, t)$ as the intensity function of a scan superimposed by two shots having intensity functions $S_1(x, y, t)$ with $t < t_2$ and $S_2(x, y, t)$ with $t > t_1$ respectively, we can model dissolve as

$$\mathrm{Dissolve}(x, y, t) = (1 - \alpha(t))S_1(x, y, t) + \alpha(t)S_2(x, y, t) \qquad t_1 < t < t_2 \tag{21}$$

where $\alpha(t) = (t - t_1)/(t_2 - t_1)$ varys linearly with t in the range [0, 1]. Denote $\mu(t)$ be the mean intensity of a scan during the interval $t_1 < t < t_2$, then we can have

$$\mu(t) = \mu^{S_1}(t) + (\mu^{S_2}(t) - \mu^{S_1}(t))\alpha(t) \tag{22}$$

where $\mu^{S_j}(t)$ is the mean intensity of a scan at time t that belongs to shot j. Taking the first derivative $\mu'(t) = (d\mu_i(t))/(dt)$, we have

$$\mu'(t) = \frac{\mu^{S_2}(t) - \mu^{S_1}(t)}{t_2 - t_1}. \tag{23}$$

Assuming $\mu^{S_1}(t)$ and $\mu^{S_2}(t)$ remain unchanged during dissolves, $\mu'(t)$ is a constant value.

Similarly, let $\sigma(t)$ be the variance of a scan during a dissolve, then

$$\sigma(t) = (\sigma^{S_1}(t) + \sigma^{S_2}(t))\alpha^2(t) - 2\sigma^{S_1}(t)\alpha(t) + \sigma^{S_1}(t) \tag{24}$$

where $\sigma^{S_j}(t)$ is the variance of a scan that belongs to shot j. If $\sigma^{S_1}(t)$ and $\sigma^{S_2}(t)$ remain constant, $\sigma(t)$ is a concave upward parabola during $t_1 < t < t_2$.

Fade-in and fade-out are treated as special cases of dissolve, either $S_1(x, y, t)$ or $S_2(x, y, t)$ will be replaced by a constant image C (black image in most cases). For fade-in, (21) becomes

$$\mathrm{FadeIn}(x, y, t) = (1 - \alpha(t))C + \alpha(t)S_2(x, y, t), \qquad t_1 < t < t_2. \tag{25}$$

Similarly, for fade out, (21) becomes

$$\mathrm{FadeOut}(x, y, t) = (1 - \alpha(t))S_1(x, y, t) + \alpha(t)C, \qquad t_1 < t < t_2. \tag{26}$$

$\mu'(t)$ remains relatively constant during fade-in and fade-out, while $\sigma(t)$ becomes a semi-parabolic curve. In addition, there are abrupt changes in scans at the beginning of fade-in and at the ending of fade-out. The abrupt changes can be detected by the cut detection algorithm described in the previous section.

Based on the above discussion, dissolves can be detected by looking for periods whose mean derivative and variance behave as (23) and (24). In the implementation, our approach detects

[8] The values 0.05% and 10% are empirically set. These values are set to be low to avoid missed detections, since most of the false alarms have been pruned by color histogram at this stage.

[9] This is because rigid object translation or camera panning (tilting) will usually generate multiple feature lines in temporal slices.

TABLE I
CUT DETECTION ON THE MOVIE SACRIFICE.mpg OF 738 FRAMES (FIVE CAMERA CUTS)

Approach	D	F	M
Slice Coherency	5	0	0
Color Histogram	5	0	0
Frame Difference	5	0	0
Step Variable	4	0	1

TABLE II
CUT DETECTION ON THE TV STREAMS TOWEST.mpg OF 18 954 FRAMES (97 CAMERA CUTS)

Approach	D	F	M
Slice Coherency	96	16	1
Color Histogram	91	13	6
Frame Difference	97	20	0
Step Variable	95	31	2

TABLE III
CUT DETECTION ON THE MOVIE SHUSHAN.mpg OF 9150 FRAMES (119 CAMERA CUTS)

Approach	D	F	M
Slice Coherency	117	34	2
Color Histogram	102	21	17
Frame Difference	117	67	2
Step Variable	113	52	6

TABLE IV
CUT DETECTION ON THE NEWS PEARL.mpg OF 4300 FRAMES (30 CAMERA CUTS)

Approach	D	F	M
Slice Coherency	30	1	0
Color Histogram	29	0	1
Frame Difference	30	1	0
Step Variable	29	4	1

TABLE V
CUT DETECTION ON THE MOVIE TUNGNIEN.mpg OF 11 247 FRAMES (17 CAMERA CUTS)

Approach	D	F	M
Slice Coherency	17	4	0
Color Histogram	17	0	0
Frame Difference	17	5	0
Step Variable	17	0	0

TABLE VI
RECALL AND PRECISION MEASURES FOR THE FIVE TESTED VIDEO

Approach	*Recall*	*Precision*
Slice Coherency	0.99	0.83
Color Histogram	0.91	0.88
Frame Difference	0.99	0.75
Step Variable	0.96	0.75

a period ($[t_1, t_2]$ and $15 \leq t_2 - t_1 \leq 45$) that has an approximate constant mean derivative, and has a semi upward parabola curve of variance in any two spatio-temporal images as a dissolve. In principle, this statistical-based approach can only be tolerant to dissolves whose $\mu^{S_1}(t)$, $\mu^{S_2}(t)$, $\sigma^{S_1}(t)$ and $\sigma_i^{S_2}(t)$ are constant over $t_1 < t < t_2$. For dissolves with motions, these statistical measures will not be constant; however, they can still demonstrate the parabolic shape of variance curve and the approximate constant of mean derivative.

VII. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed approach, we conduct experiments on news sequences, documentary films, movies, and TV streams. The size of an image frame is 352×240. We first examine the performance of the cut, wipe, and dissolve detectors independently on the image sequences. We then activate all detectors on testing two image sequences to demonstrate their capability on classifying camera breaks, and their tolerance to false and missed detections. We employ recall-precision to analytically measure the results. Denote A_i as the number of frames in class i, B_i as the number of detected frames in class i, and D_i as the number of correctly detected frames in class i. Then

$$\text{recall}_i = \frac{D_i}{A_i} \tag{27}$$

$$\text{precision}_i = \frac{D_i}{B_i} \tag{28}$$

where $i \in \{\text{cut}, \text{wipe}, \text{dissolve}\}$, recall_i and precision_i are in the interval of [0, 1]. Low recall values indicate the frequent occurrence of missed detections, while low-precision values show the frequent occurrence of false alarms. For instance, if only 20 frames are detected for a dissolve of 30 frames, then $\text{precision} = 1$ and $\text{recall} = 2/3$, while if there are 40 frames detected, then $\text{precision} = 3/4$ and $\text{recall} = 1$.

For simplicity, we define N as the number of frames in a sequence; D as the number of correct detections; F as false alarms; and M as missed detections.

A. Cut Detection

We compare the performance of our proposed cut detection method (namely slice coherency) with three other approaches: color histogram [13], [17], [21], frame difference [17], [21], and step variable [18]. The first two approaches work directly on the

TABLE VII
DETECTION RESULTS ON VARIOUS WIPE PATTERNS

Wipe Pattern	Actual Wipe Range	Detected Wipe Range	
		Slice Coherency	Statistical Approach
Iris round	80-109	79-103	none
Iris shape	80-109	88-111	none
Radial wipe	80-109	83-106	none
Clock wipe	80-109	86-106	none
Flip over	80-109	80-109	80-109
Spiral box	80-109	84-110	none
Zig-zag block	80-109	81-107	none
Checker wipe	80-109	82-95	none
Motion wipe	80-109	86-106	101-109
Zoom wipe	43-83	47-73	none
Barn door	34-63	32-57	34-72 82-89
Band wipe	31-45	29-44	31-93
Wipe	41-63	41-62	40-69 77-81
Page turn	54-73	55-65	61-66

dc sequence of MPEG videos, while the last approach works in the uncompressed image domain. The color histogram difference between two dc images f_1 and f_2 are

$$\mathrm{HD}(f_1, f_2) = \sum_{k=1}^{64} \{|h_1^r(k) - h_2^r(k)| + |h_1^g(k) - h_2^g(k)| + \left|h_1^b(k) - h_2^b(k)\right|\} \quad (29)$$

where h_i^r, h_i^g and h_i^b are the histograms corresponding to the RGB components of a dc image f_i. The histogram is set to 64 bins since it has been shown to give sufficient accuracy [21]. The frame difference is computed as

$$\mathrm{FD}(f_1, f_2) = \sum_i \sum_j |f_1(i,j) - f_2(i,j)|. \quad (30)$$

Since the dc image is inherently smooth, it is less sensitive to camera and object motion compared to the full frame's pixel-level difference [17]. In the implementation, a camera cut is detected if the corresponding difference is a local maximal.

In contrast to other approaches operated in the compressed domain, step-variable [18] speeds up the processing time by subsampling video frames. Along the temporal dimension, two frames separated by t time units are compared; along the spatial dimension, only the predefined set of blocks in a frame are compared. The value of $t = \{1, 2, 4, 8, 16, \ldots\}$ is set adaptively based on the scene activities. The mean value difference of each block in two compared frames is used to detect cuts.

Tables I–V show detection results of the four different approaches for five tested sequences. In Table I, the video sacrifice.mpg consists of five shots taken in a scene. Non-adjacent shots may have similar color-texture properties, as a result step-variable misses one of the shots. In Table II, the video towest.mpg has rich fighting and magical scenes, as a result, false alarms and missed detections are arisen by all approaches. Similarly, the shushan.mpg in Table III, which has rich special cinematographical effects, also causes the same problems. In Table IV, the pearl.mpg is a news sequence with some sport scenes with fast and large object motions. Although the color histogram does not cause any false alarm, a miss has happened at the location where there are two adjacent shots of a soccer field taken from two different view points. In Table V, the movie tungnien.mpg consists of both indoor and outdoor long-take shots.[10] In one of the shots, there is an object moving in and out of the screen abruptly; as a result, both the slice coherency and frame difference approaches give rise to false alarms.

Table VI shows the recall-precision measures of all approaches for the five tested video. While frame difference shares with slice coherency the best recall rate, it suffers from the lowest precision rate. On the other extreme, color histogram has the highest precision but the lowest recall rate. The results are not surprising since frame difference can only model local changes while color histogram can only handle global changes. Our proposed approach acquires a better trade-off in terms of recall and precision mainly due to the presence of coherency which provides useful information for cut detection. Although slice coherency only processes partial information as the strategies adopted by step-variable, it is comparatively tolerant to both missed and false detections.

B. Wipe Detection

We compare the performance of our wipe detection algorithm (namely slice coherency) with the statistical approach proposed

[10] Stationary camera with or without object motions.

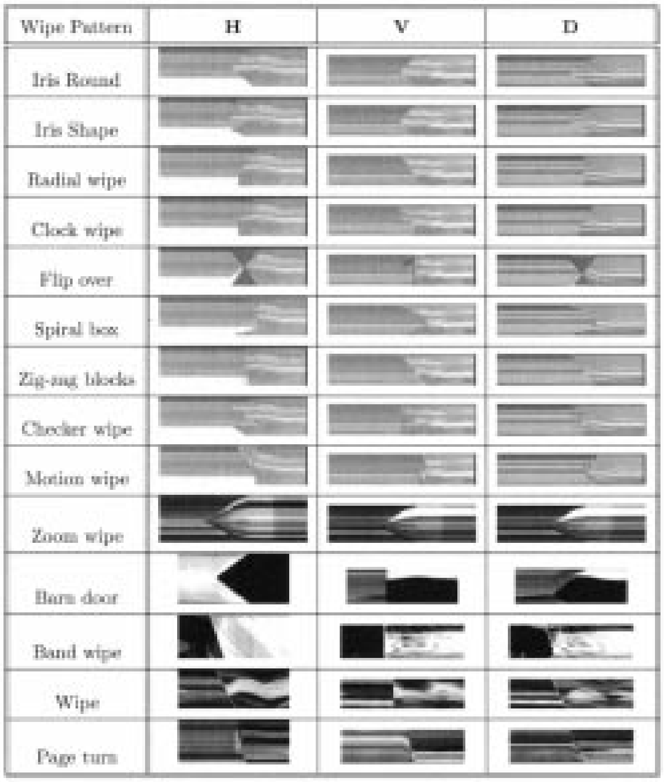


Fig. 12. Spatio-temporal images of various wipe patterns.

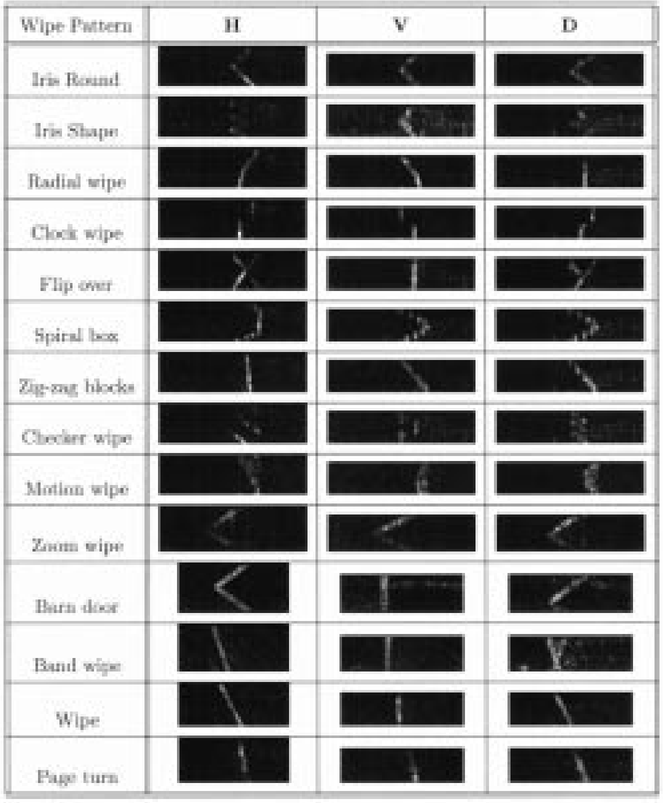


Fig. 13. Computed energy for the spatio-temporal slices of various wipe patterns.

by Alattar [2]. The statistical approach first detects the spikes in the second derivative of the mean and variance of frames. These spikes mark the end or the start and end of a potential wipe. The approach then investigates the average values of the first derivative of the mean and variance of a suspected wipe region. A wipe is detected if the average value is above a threshold.

We test these two algorithms with 14 different wipe transitions which are shown in Fig. 11. All the tested videos consist of two shots with slight to moderate motions. Table VII lists the wipe transitions along the actual and detected wipe frames by these two algorithms. Our proposed algorithm shows a significantly better performance than the statistical approach in terms of recall and precision, as listed in Table VIII. The statistical approach fails in detecting some wipes because of the absence of sharp spikes in the wipe regions. Moreover, it is blind when marking the boundary of a wipe if there are motions in two shots. In contrast, our approach successfully detect all wipes except that few frames at the boundary of some wipes are missed or over-estimated. We also test the cut detectors (color histogram, frame difference and step-variable) as discussed in Section VII-A on the 14 wipe transitions. However, none of the wiped frame is detected by these three approaches since the difference between two adjacent wiped frames is small.

Fig. 12 shows the spatio-temporal slices created by the wipe transitions in Table VII, while Fig. 13 shows the computed energy U_{wipe} of these spatio-temporal slices. It is worth noticing that the regional boundaries of the three selected spatio-temporal slices ($\mathbf{H}$, $\mathbf{V}$, and $\mathbf{D}$) cover most of the wipe transition periods since most wipes start at one direction/corner and end at the opposite direction/corner (e.g., wipe and spiral box), or start at the center and grow outward (e.g., zoom wipe and barn door wipe). Among the 14 tested wipe transitions, only four frames in clock wipe, eight frames in motion wipe, and three frames in page turn are not covered by the regional boundaries of the three selected spatio-temporal slices.

C. Dissolve Detection

Table IX shows the experimental results of our proposed method on two tested videos. All dissolves involve slight motions and cross about 30 frames. For the detected dissolve sequences, the number of undetected dissolve frames on average are six frames. Fig. 14 illustrates an example of how dissolves are detected. The missed detections are due to the low value of variance between two shots; as a result, the shape of the parabolic cannot be detected. The recall and precision values of the two tested videos are 0.90 and 0.88, respectively.

D. Camera Break Detection

In this section, we integrate the cut, wipe, and dissolve detection algorithms to detect camera breaks. The dissolve detection is started after all the cuts and wipes are detected. The experimental results are summarized in Table X. Most of the detected cut and wipe frames are classified correctly. The two false alarms in cut detection are due to the sharp change of illumination. The only false alarm arisen in wipe detection is because of a large object that moves across the screen from bottom to top. The two missed wipes are due to the low contrast between two connected shots and a long wiped period (about 90 frames).

TABLE IX
DISSOLVE DETECTION RESULTS

Video	Frames	D	F	M
Xpearl.mpg	2000	11	0	1
XShuShan.mpg	2280	16	0	1

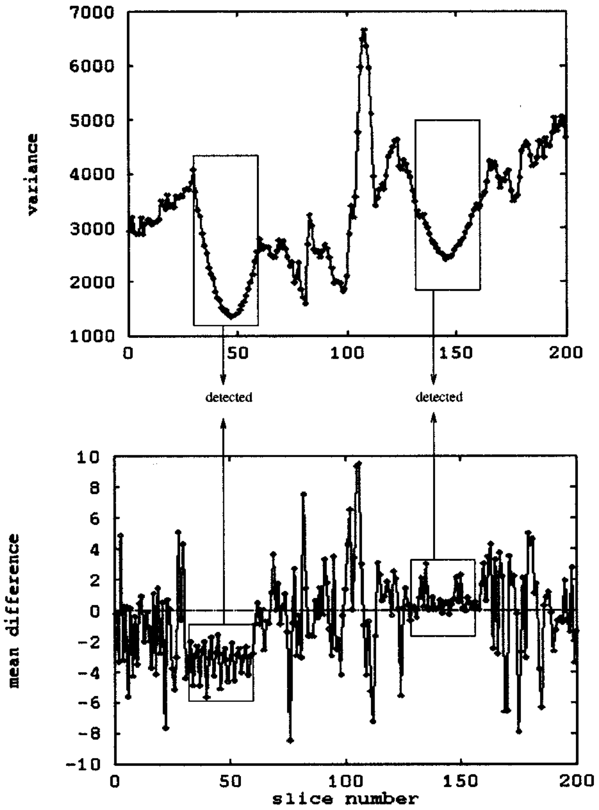


Fig. 14. Detection of dissolves by looking for the parabolic curves of variance and the approximate constant of mean derivative in a horizontal spatio-temporal slice.

TABLE X
CAMERA BREAK DETECTION RESULTS

Tested	Total	Cut			Wipe			Dissolve		
Video	Frames	D	F	M	D	F	M	D	F	M
ba.mpg	10170	46	0	0	2	0	0	4	2	2
gf.mpg	10000	54	0	0	4	0	0	4	2	0
ha.mpg	15420	53	2	0	6	1	2	23	2	6
recall		1.00			0.75			0.76		
precision		0.99			0.80			0.77		

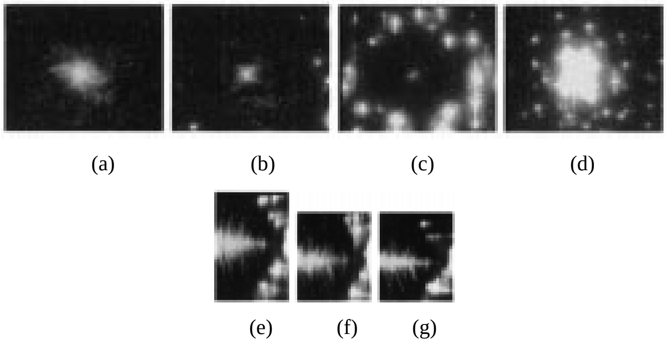


Fig. 15. A shot of 30 frames (670th to 699th). Sample image frames: (a) 680th; (b) 685th; (c) 690th; and (d) 699th. (e) Horizontal spatio-temporal slice. (f) Vertical spatio-temporal slice. (g) Diagonal spatio-temporal slice.

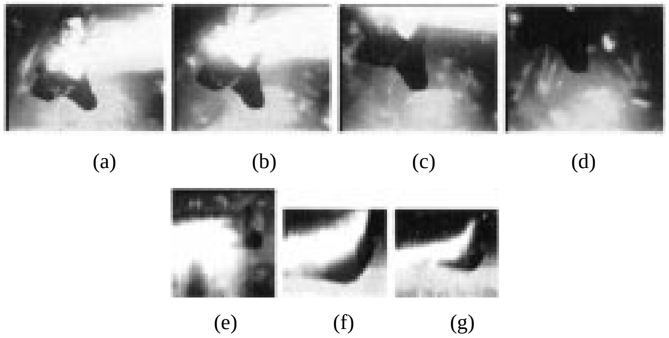


Fig. 16. A shot of 42 frames (8894th to 8935th). Sample image frames: (a) 8916th; (b) 8923rd; (c) 8928th; and (d) 8931st. (e) Horizontal spatio-temporal slice. (f) Vertical spatio-temporal slice. (g) Diagonal spatio-temporal slice.

Among the correctly detected wipes, the average number of over detected wiped frames is five, and the number of missed detected wiped frames is two. The missed dissolves are due to the long dissolve period (120 frames), and object motions during dissolve periods. False dissolve detections are due to camera motions which cause the mean derivative and variance of the corresponding scans resemble a dissolve pattern.

VIII. DISCUSSION

To analyze the pros and cons of our proposed approach, we summarize four main observations found in the experiments.

1) *Presence of structural information*: Compared with other approaches, slice coherency can handle fast motions and color changes within a shot. This is due to the presence of structural information provided by the regional boundaries in spatio-temporal slices. The structural information is not only exploited to classify cut, wipe and dissolve, but is also employed to distinguish wipe, motion, and color changes. For instance, Fig. 15 shows a shot undergoing significant changes of color, while other approaches raise false alarms, slice coherency successfully classifies it as one shot. Figs. 16 and 17 further show two shots of fast motion which have given rise to false alarms by other approaches, but however, induce no error by our proposed method.

2) *Presence of local information*: Unlike the color histogram, slice coherency can also capture local changes. In Fig. 18, the color histogram fails to detect the cut due to the similar color distribution between two adjacent shots; however, slice coherency succeeds as there is a shift of spatial texture arrangements. Compared with frame difference which can also detect the cut in Fig. 18, our approach is more efficient since only partial information of dc images are used. A step variable, which also

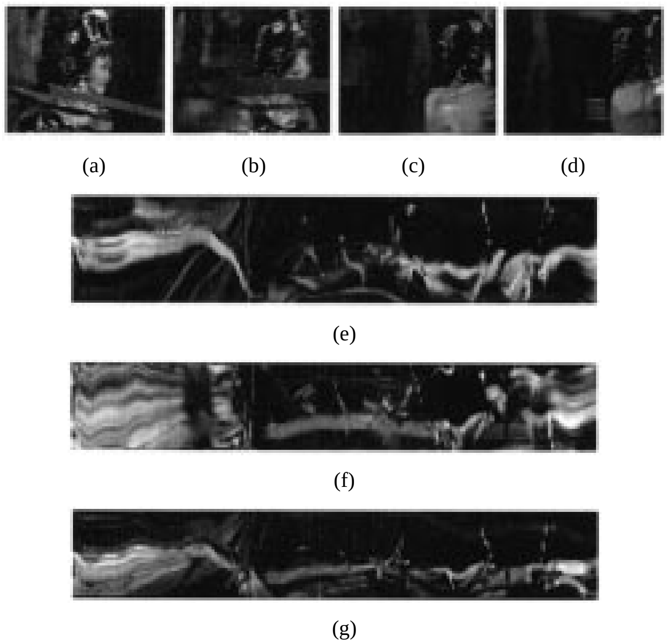


Fig. 17. A shot of 216 frames (3263th to 3478th). Sample image frames: (a) 3330th; (b) 3334th; (c) 3338th; and (d) 3342nd. (e) Horizontal spatio-temporal slice. (f) Vertical spatio-temporal slice. (g) Diagonal spatio-temporal slice.

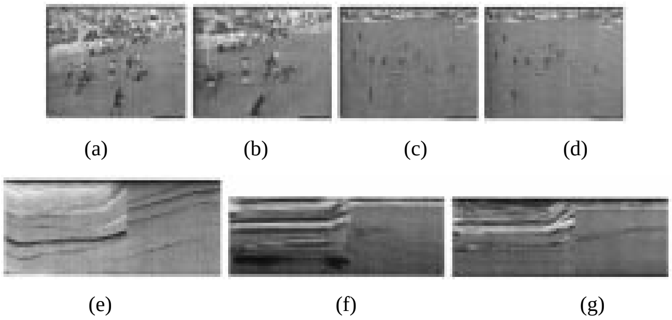


Fig. 18. Two shots of 100 frames (3600th to 3657th and 3658th to 3699th). Sample image frames: (a) 3650th; (b) 3657th; (c) 3658th; and (d) 3660th. (e) Horizontal spatio-temporal slice. (f) Vertical spatio-temporal slice. (g) Diagonal spatio-temporal slice.

processes partial information, however, fails in detecting the cut.

3) *Problems in dissolve detection*: The statistical properties discussed in Section VI are not unique to dissolve. Camera and object motions can have similar statistical patterns. This is the main challenge for the general success of dissolve detection. In addition, the intervention of motion during the dissolve period will perturb the statistical properties of a dissolve. As a result, the detection of an exact dissolve period is almost impossible. The duration of a dissolves can vary from one to hundred frames. A dissolve of long period is extremely difficult to be detected since the statistical change between two adjacent shots is not easily seen.
4) *Problems remain for all approaches*: Sharp illumination changes are still difficult problems which remain to be solved. Fig. 19 illustrates an example where the illumination effect creates two distinct regions in the spatio-temporal slices.

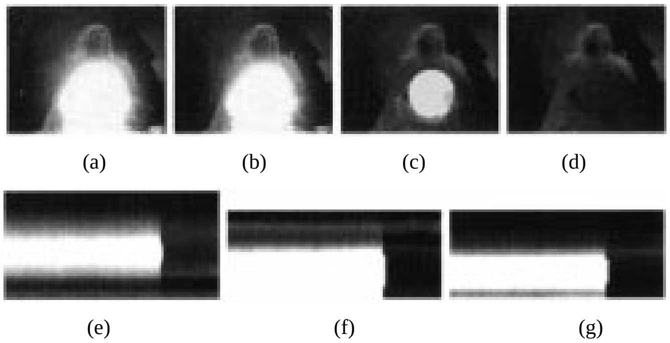


Fig. 19. A shot of 88 frames (2600th to 2687th). Sample image frames: (a) 2660th; (b) 2663rd; (c) 2664th; and (d) 2670th. (e) Horizontal spatio-temporal slice. (f) Vertical spatio-temporal slice. (g) Diagonal spatio-temporal slice.

In terms of speed efficiency, our proposed approach operates in real-time (30 frames/s). On a Pentium II platform, our camera break detection algorithm (include cut, wipe and dissolve detections) runs in the speed of 35 frames/s. Among the three detectors, cut detector operates in 40 frames/s, wipe detector in 38 frames/s, and dissolve detector in 86 frames/s. The most time-consuming part in the algorithm is the convolution of Gaussian and Gabor filters with spatio-temporal slices, as discussed in Section III-A.

IX. Conclusion

We have presented a procedure for detecting and classifying cuts, wipes, and dissolves based on the analysis of spatio-temporal slices. Our approach reduces video-segmentation problems to image-segmentation problems, and in addition, processes frames directly in the MPEG domain, resulting in an efficient framework. The proposed algorithms can compromise the recall and precision performace of cut detection, handle various types of wipe transitions, and detect most linear dissolves, even though only partial data is analyzed. In the future, we will study a more sophisticated dissolve detection algorithm and the possibility of estimating image and motion features directly from the rhythm of shots for video database indexing and retrieval.

References

[1] E. H. Adelson and J. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer.*, vol. 2, no. 2, pp. 284–299, Feb. 1985.

[2] A. M. Alattar, "Wipe scene change detector for use with video compression algorithms and MPEG-7," *IEEE Trans. Consumer Electron.*, vol. 44, pp. 43–51, Feb. 1998.

[3] ——, "Detecting and compressing dissolve regions in video sequences with a DVI multimedia image compression algorithm," in *IEEE Int. Symp. Circuits and Systems*, vol. 1, 1993, pp. 13–16.

[4] P. J. Burt, "Fast filter transform for image processing," *Comput. Graphics, Image Processing*, vol. 16, pp. 20–51, 1981.

[5] D. Bordwell and K. Thompson, *Film Art: An Introduction*, 2nd ed. New York: Random House, 1986.

[6] A. M. Ferman and A. M. Tekalp, "Efficient filtering and clustering methods for temporal video segmentation and visual summarization," *J. Vis. Commun. Image Represent.*, vol. 9, no. 4, pp. 336–351, 1998.

[7] A. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognit.*, vol. 24, no. 12, 1991.

[8] S. Z. Li, *Markov Random Field Modeling in Computer Vision*. New York: Springer-Verlag, 1995.

[9] H.-C. H. Liu and G. L. Zick, “Scene decomposition of MPEG compressed video,” in *Proc. SPIE Conf. Digital Video Compression: Algorithms and Technologies*, vol. 2419, 1995, pp. 26–37.

[10] J. Meng, Y. Juan, and S. F. Chang, “Scene change detection in a MPEG compressed video sequence,” in *Proc. SPIE Conf. Digital Video Compression: Algorithms and Technologies*, vol. 2419, 1995, pp. 14–25.

[11] C. W. Ngo, T. C. Pong, and R. T. Chin, “Camera breaks detection by partitioning of 2D spatio-temporal images in MPEG domain,” *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, vol. 1, pp. 750–755, 1999.

[12] ——, “Detection of gradual transitions through temporal slice analysis,” *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 36–41, 1999.

[13] N. V. Patel and I. K. Sethi, “Compressed video processing for cut detection,” in *IEE Proc. Visual Image Signal Processing*, vol. 143, Oct. 1996, pp. 315–323.

[14] S. L. Peng, “Temporal slice analysis of image sequences,” *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 283–288, 1991.

[15] R. O. Duda and P. E. Hart, “Use of Hough transformation to detect lines and curves in pictures,” *Commun. ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[16] M. Wu, W. Wolf, and B. Liu, “An algorithm for wipe detection,” in *IEEE Int. Conf. Image Processing*, vol. 1, 1998, pp. 893–897.

[17] B. L. Yeo and B. Liu, “On the extraction of DC sequence from MPEG compressed video,” in *IEEE Int. Conf. Image Processing*, vol. 2, Oct. 1995, pp. 260–3O.

[18] W. Xiong and C. M. Lee, “Efficient scene change detection and camera motion annotation for video classification,” *J. Comput. Vis. Image Understanding*, vol. 17, no. 2, pp. 161–181, 1998.

[19] B. L. Yeo and B. Liu, “Rapid scene analysis on compressed video,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 533–544, Dec. 1995.

[20] H. M. Yip and T. C. Pong, “Detection of moving objects using a spatiotemporal representation,” in *Proc. Int. Conf. Pattern Recognition*, vol. 1, 1996, pp. 483–487.

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

Chong-Wah Ngo received the B.Sc. and M.Sc. degrees, both in computer engineering, from Nanyang Technological University of Singapore in 1994 and 1996, respectively, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology (HKUST) in 2000.

He is currently a Research Associate at HKUST. Since April 2001, he has also been a post-doctoral Visitor with the Beckman Institute, University of Illinois at Urbana–Champaign. He was with Information Technology Institute, Singapore, during 1996, and with Microsoft Research China as a summer intern in 1999. His current research interests include image and video indexing, computer vision, and pattern recognition.

Ting-Chuen Pong received the Ph.D. degree in computer science from Virginia Polytechnic Institute and State University, Blacksburg, in 1984.

In 1991, he joined the Hong Kong University of Science and Technology (HKUST), where he is currently a Reader of Computer Science and Associate Dean of Engineering. Prior to joining HKUST, he was an Associate Professor in Computer Science with the University of Minnesota at Minneapolis.

Dr. Pong served as Program Co-Chair of the Third International Computer Science Conference in 1995 and the Third Asian Conference on Computer Vision in 1998. He is currently on the Editorial Board of the *Pattern Recognition Journal*. He is a recipient of the Annual Pattern Recognition Society Award in 1990 and Honorable Mention Award in 1986.

Roland T. Chin received the B.S. degree (Hons.) and the Ph.D. degree, both in electrical engineering, from the University of Missouri at Columbia.

From 1979 to 1981, he was a Researcher at NASA Goddard Space Flight Center, Greenbelt, MD. He was on the faculty of Electrical and Computer Engineering, University of Wisconsin at Madison, from 1981 to 1995. He became a Full Professor in 1989 and served as Associate Department Chair from 1986 to 1990. Since 1993, he has been on the faculty of the Computer Science Department, Hong Kong University of Science and Technology, and is now a Department Head.

Dr. Chin served on the Editorial Board of the the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and the *Asian Pacific Engineering Journal*. He was the recipient of the National Science Foundation Presidential Young Investigator Award in 1984.