International Journal of Computer Vision 50(2), 127–142, 2002

Motion-Based Video Representation for Scene Change Detection

CHONG-WAH NGO
Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, People's Republic of China
cwngo@cs.cityu.edu.hk

TING-CHUEN PONG
Department of Computer Science, The Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong, People's Republic of China
tcpong@cs.ust.hk

HONG-JIANG ZHANG
Microsoft Research Asia, 5/F Beijing Sigma Center, No. 49, ZhiChun Road, Haidian District, Beijing 100080, People's Republic of China
hjzhang@microsoft.com

Received March 16, 2001; Revised August 15, 2001; Accepted December 31, 2001

Abstract. In this paper, we present a new framework to automatically group similar shots into one scene, where a scene is generally referred to as a group of shots taken place in the same site. Two major components in this framework are based on the motion characterization and background segmentation. The former component leads to an effective video representation scheme by adaptively selecting and forming keyframes. The later is considered novel in that background reconstruction is incorporated into the detection of scene change. These two components, combined with the color histogram intersection, establish our basic concept on assessing the similarity of scenes.

Keywords: scene change detection, spatio-temporal slice, keyframe formation, background reconstruction

1. Introduction

A video usually consists of scenes, and each scene includes one or more shots. A shot is an uninterrupted segment of video frame sequence with static or continuous camera motion, while a scene is a series of consecutive shots that are coherent from the narrative point of view. These shots are either shot in the same place or they share similar thematic content. By decomposing a video into scenes, we can facilitate content-based video browsing and summary. Figure 1 depicts the structural content of a typical video. The goal of this paper is to propose a framework for structuring the content of videos in a bottom-up manner, as illustrated in Fig. 1, while abstracting the main content from video frames.

1.1. Challenge

Intuitively, scene change detection can be tackled from two aspects: comparing the similarity of background scenes in shots and analyzing the content of audio features. Nevertheless, there are several research problems along this thought: (i) background and foreground segmentation; (ii) background and foreground identification; (iii) similarity measure; and (iv) word spotting from audio signal. The first problem can be

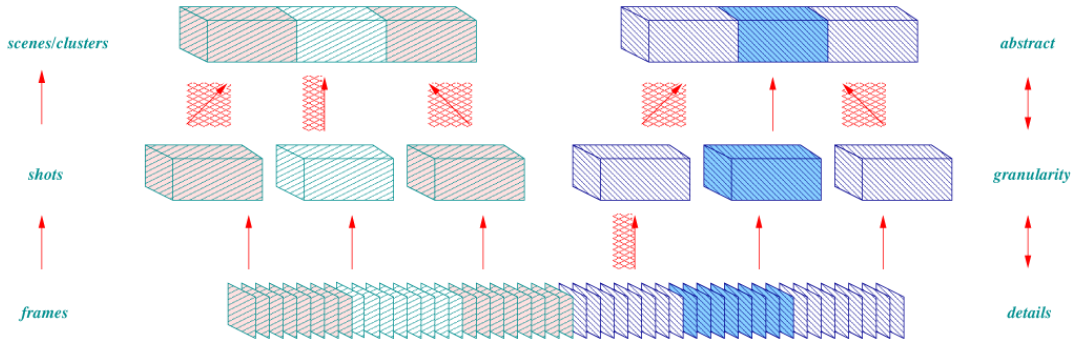


Figure 1. Video structure.

solved satisfactorily only when the background and foreground objects have different motion patterns. The second problem requires high-level knowledge and, in most cases, necessitates manual feedback from human. The third problem has been addressed seriously since the beginning of content-based image and video retrieval (Flickner et al., 1995; Gudivada and Raghavan, 1995) research. A good piece of work on similarity matching can be found in Jacobs et al. (2000) and Santini and Jain (1999). The last problem is still regarded as hard since video soundtracks are complex and often mixed with many sound sources.

Scene change detection, in general, is considered a difficult task based on the problems discussed above. A fully automatic system cannot be easily realized. Since a complete and reliable segmentation cannot be done prior to the detection of a scene, shot representation and similarity measure need to be reconsidered, in order to automate this process.

1.2. Related Works

Previous work on scene change detection includes (Corridoni and Del. Bimbo, 1998; Hanjalic et al., 1999; Sundaram and Chang, 2000; Huang et al., 1998; Rui et al., 1998; Sahouria and Zakhor, 1999; Yeung and Yeo, 1997). Basically there are two major approaches: one adopts the time-constraint clustering algorithm to group shots which are visually similar and temporally closed as a scene (Corridoni and Del. Bimbo, 1998; Hanjalic et al., 1999; Rui et al., 1998; Sahouria and Zakhor, 1999; Yeung and Yeo, 1997); the other employs audiovisual characteristics to detect scene boundaries (Sundaram and Chang, 2000; Huang et al., 1998). In general, the success of these approaches relies on the video representation scheme and shot similarity measure. The former aims at representing a video in a compact yet semantically meaningful way, while the later attempts to mimic human perception capability. In most systems, shots are represented by a set of selected keyframes, and the similarities among the shots are soly or partially[1] dependent on the color similarity of those keyframes (Corridoni and Del. Bimbo, 1998; Hanjalic et al., 1999; Rui et al., 1998; Yeung and Yeo, 1997).

In this paper, we propose a motion-based video representation scheme for scene change detection, by integrating our previous works on video partitioning (Ngo et al., 1999, 2000a, 2001), motion characterization (Ngo et al., 2000b) and foreground vs background segmentation (Ngo et al., 2000b; Ngo, 2000). We tackle the problem from four different aspects: (i) represent shots adaptively and compactly through motion characterization; (ii) reconstruct background in the multiple motion case; (iii) reduce the distraction of foreground objects by histogram intersection (Swain and Ballard, 1991); and (iv) impose time-constraint to group shots that are temporally closed. Compared with Corridoni and Del. Bimbo (1998), Hanjalic et al. (1999), Huang et al. (1998), Rui et al. (1998), Sahouria and Zakhor (1999), and Yeung and Yeo (1997), aspects (i), (ii) and (iii) are considered new features to scene change detection. The issue of compact video representation for shot similarity measure has not yet been fully addressed by previous approaches. For instance, the approach in (Hanjalic et al., 1999) simply selects a few image frames as keyframes for similarity measure. The similarity of two shots is computed to be the color similarity of two image frames, which may consequently lead to the occurrence of missed detections. In contrast, our approach not only selects keyframes from shots, but also

reconstruct new images such as background panoramas as new keyframes based on the annotated motion of shots. Since the proposed video representation scheme is compact, the histogram intersection which measures similarity between features based on the intersection of feature points, can be more effectively performed for scene change detection.

2. Framework

Figure 2 depicts the basic framework of our scene change detection approach. An input video is first partitioned into shots. Those shots that have more than one camera motion are temporally segmented into motion coherent sub-units, and each sub-unit is characterized according to its camera motion. A test is then conducted to check if a sub-unit has more than one motion (e.g., both camera and object motion). For multiple motion cases, the corresponding sub-units are further spatially decomposed into motion layers. The dominant motion layer of a sub-unit is subsequently reconstructed to form a background image. For other cases, keyframe selection and formation are adaptively performed based on the annotated motion to compactly represent the content of a shot. Finally, scene change is detected by grouping shots with similar color content.

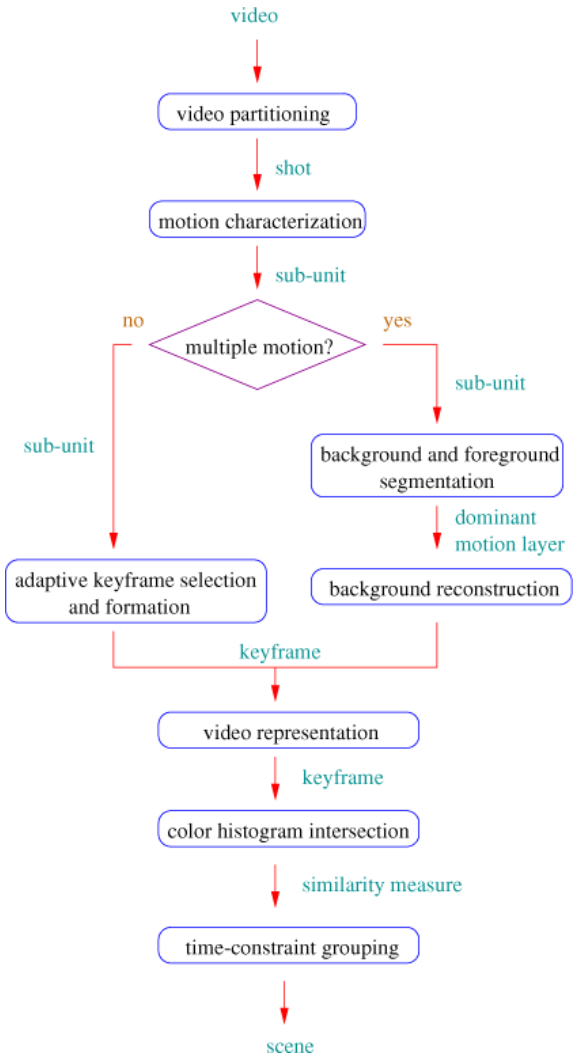


Figure 2. A scheme for scene change detection.

Our works on video partitioning, motion characterization, and background vs foreground segmentation are based on the pattern analysis and processing of spatio-temporal slices (STS). In this paper, we will only concentrate on the approaches for video representation, similarity measure, and time-constraint grouping which basically take the computed results of STS pattern analysis as input. A brief introduction to STS pattern analysis is given in the next section.

3. Processing of Spatio-Temporal Slices (STS)

If we view a video as an image volumn with (x, y) image dimension and t temporal dimension, the spatio-temporal slices are a set of $2D$ images in a volumn with one dimension in t, and the other in x or y. One example is given in Fig. 3; the horizontal axis is t, while the vertical axis is x. For our application, we process all slices, both horizontal and vertical, in a volume to analyze the spatio-temporal patterns due to various motions. For simplicity, we denote horizontal slices as **H** with dimension (x, t), and vertical slices as **V** with dimension (y, t).

A spatio-temporal slice, by first impression, is composed of color and texture components. On one hand, the discontinuity of color and texture represents the occurrence of a new event; on the other hand, the orientation of texture depicts camera and object motions. While traditional computer vision and image processing literature tend to formulate methodologies on two adjacent frames, spatio-temporal slices, in a complementary way, provide rich visual cues along a larger temporal scale for video processing and representation. The former gives a snapshot of motion field; the later, in contrast, offers a glance of motion events.

Figure 3 shows a spatio-temporal slice extracted from the center row of a video composed of six shots.

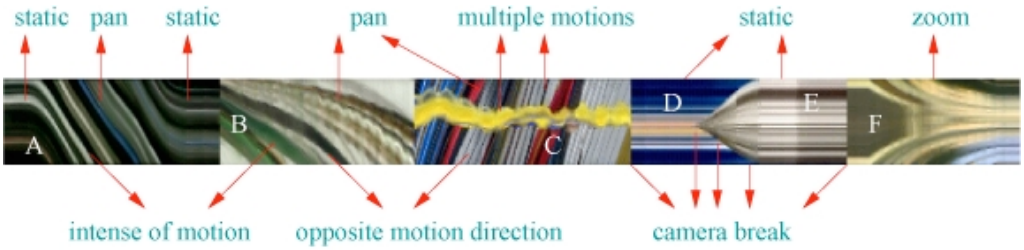


Figure 3. Patterns in a spatio-temporal slice.

By careful observation of the patterns inherent in this slice, it is not difficult to perceive the following cues:

- *Shot boundary* which is located at a place where the color and texture in a slice show a dramatic change. This change may involve more than one frame as indicated by the boundary of shots D and E.
- *Camera motion* is inferred directly from the texture pattern. For instance, horizontal lines depict stationary camera and object motion; slanted lines depict camera panning.[2] In addition, the orientation of slanted lines represent motion direction (in shot B, the camera moves to the left; in shot C, the camera moves to the right), while the gradient of slanted lines is proportional to motion intensity (the speed of panning in shot A is faster than shot B). Based on this observation, it is simple to find that shot A is composed of different camera motions. In this case, shot A can be temporally segmented into three sub-units.
- *Multiple motions* are perceived when two dissimilar texture patterns appear in a shot, as shown in shot C. In this shot, the yellow region describes a non-rigid object motion, while the background region indicates camera panning.

In our approach, shot boundaries are detected by color and texture segmentation (video partitioning) (Ngo et al., 1999, 2000a, 2001); the motion information is estimated through the orientation and gradient of line patterns (motion characterization) (Ngo et al., 2000b); motion layers are obtained by decomposing dissimilar color and texture regions in the spatio-temporal slices of a shot (background and foreground segmentation) (Ngo et al., 2000b; Ngo, 2000).

3.1. Computational Issue

For computational and storage efficiency, we propose to process and analyze spatio-temporal slices directly in the compressed video domain (MPEG domain). Slices can be obtained from the DC image[3] volume which is easily constructed by extracting the DC components[4] of MPEG video. The resulting data is smoothed while the amount is reduced by 64 times in the MPEG domain. For an image of size $M \times N$, the dimension of the corresponding DC image is $\frac{M}{8} \times \frac{N}{8}$. For a shot with T frames, the dimension of spatio-temporal slices are reduced from $M \times T$ to $\frac{M}{8} \times T$ (or $N \times T$ to $\frac{N}{8} \times T$) in the compressed domain. Hence, given a video composed of K shots, the number of slices extracted for processing are $K \times \frac{M+N}{8}$.

3.2. Motion Analysis of STS Patterns

Our approach is based on the structure tensor computation introduced in Jähne (1991) to estimate the local orientations of a slice. By investigating the distribution of orientations in all slices, we can classify motion types as well as separate different motion layers.

3.2.1. Structure Tensor. The tensor Γ of a slice[5] $\mathbf{H}$ can be expressed as

$$\Gamma = \begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xt} \\ \mathbf{J}_{xt} & \mathbf{J}_{tt} \end{bmatrix} = \begin{bmatrix} \sum_w \mathbf{H}_x^2 & \sum_w \mathbf{H}_x \mathbf{H}_t \\ \sum_w \mathbf{H}_x \mathbf{H}_t & \sum_w \mathbf{H}_t^2 \end{bmatrix} \quad (1)$$

where $\mathbf{H}_x$ and $\mathbf{H}_t$ are partial derivatives along the spatial and temporal dimensions respectively. The window of support w is set to 3×3 throughout the experiments. The rotation angle θ of Γ indicates the direction of a gray level change in w. Rotating the principle axes of Γ by θ, we have

$$\mathbf{R} \begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xt} \\ \mathbf{J}_{xt} & \mathbf{J}_{tt} \end{bmatrix} \mathbf{R}^T = \begin{bmatrix} \lambda_x & 0 \\ 0 & \lambda_t \end{bmatrix} \quad (2)$$

where

$$\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

From (2), since we have three equations with three unknowns, θ can be solved and expressed as

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\mathbf{J}_{xt}}{\mathbf{J}_{xx} - \mathbf{J}_{tt}} \tag{3}$$

The local orientation ϕ of a w in slices is computed as

$$\phi = \begin{cases} \theta - \dfrac{\pi}{2} & \theta > 0 \\ \theta + \dfrac{\pi}{2} & \text{otherwise} \end{cases} \qquad \phi = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{4}$$

It is useful to introduce a certainty measure to describe how well ϕ approximates the local orientation of w. The certainty c is estimated as

$$c = \frac{(\mathbf{J}_{xx} - \mathbf{J}_{tt})^2 + 4\mathbf{J}_{xt}^2}{(\mathbf{J}_{xx} + \mathbf{J}_{tt})^2} = \left(\frac{\lambda_x - \lambda_t}{\lambda_x + \lambda_t}\right)^2 \tag{5}$$

and $c = [0, 1]$. For an ideal local orientation, $c = 1$ when either $\lambda_x = 0$ or $\lambda_t = 0$. For an isotropic structure i.e., $\lambda_x = \lambda_t$, $c = 0$.

3.2.2. Tensor Histogram. The distribution of local orientations across time inherently reflects the motion trajectories in an image volume. A $2D$ tensor histogram $\mathbf{M}(\phi, t)$, with an 1D orientation histogram as the first dimension and time as the second dimension, can be constructed to model the distribution. Mathematically, the histogram can be expressed as

$$\mathbf{M}(\phi, t) = \sum_{\Omega(\phi,t)} c(\Omega) \tag{6}$$

where $\Omega(\phi, t) = \{\mathbf{H}(x, t) \mid \Gamma(x, t) = \phi\}$ which means that each pixel in slices votes for the bin (ϕ, t) with the certainty value c. The resulting histogram is associated with a confidence measure of

$$\mathbf{C} = \frac{1}{T \times M \times N}\sum_{\phi}\sum_{t}\mathbf{M}(\phi, t) \tag{7}$$

where T is the temporal duration and $M \times N$ is the image size. In principle, a histogram with low $\mathbf{C}$ should be rejected for further analysis.

Motion trajectories can be traced by tracking the histogram peaks over time. These trajectories can correspond to (i) object and/or camera motions; (ii) motion parallax with respect to different depths. Figure 4 shows two examples, in (a) one trajectory indicates the non-stationary background, and the other indicates the moving objects; in (b) the trajectories correspond to parallax motion.

3.3. Motion Characterization

Tensor histograms offer useful information for temporally segmenting and characterizing motions. Our algorithm starts by tracking a dominant trajectory along the temporal dimension. A dominant trajectory $p(t) = \max_{-\frac{\pi}{2}<\phi<\frac{\pi}{2}}\{\mathbf{M}(\phi, t)\}$ is defined to have

$$\frac{\sum_{t=k}^{k+15} p(t)}{\sum_{t=k}^{k+15}\sum_{\phi}\mathbf{M}(\phi, t)} > \tau \tag{8}$$

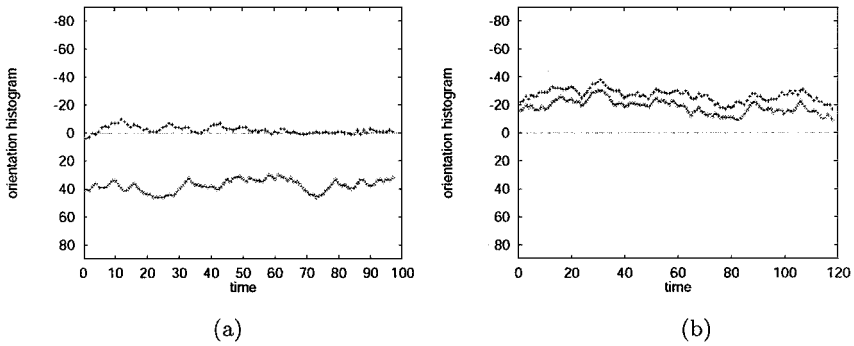


Figure 4. Motion trajectories in the tensor histograms. (a) Moving object. (b) Parallax panning.

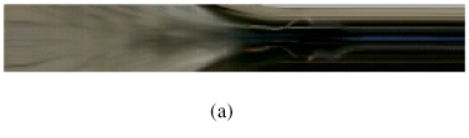
The dominant motion is expected to stay steady for approximately fifteen frames (0.5 seconds). The threshold value $\tau = 0.6$ is empirically set to tolerate camera jitter. After a dominant trajectory is detected, the algorithm simultaneously segments and classifies the dominant motion trajectory. A sequence with static or slight motion has a trajectory of $\phi = [-\phi_a, \phi_a]$. Ideally, ϕ_a should be equal to 0. The horizontal slices of a panning sequence form a trajectory at $\phi > \phi_a$ or $\phi < -\phi_a$. If $\phi < -\phi_a$, the camera pans to the right; if $\phi > \phi_a$, the camera pans to the left. A tilting sequence is similar to a panning sequence, except that the trajectory is traced in the tensor histogram generated by vertical slices. The parameter ϕ_a is empirically set to $\frac{\pi}{36}$ (or 5°) throughout the experiments. For zoom, the tensor votes are approximately symmetric at $\phi = 0$. Hence, instead of modeling as a single trajectory, the zoom[6] is detected by

$$\frac{\sum_{\phi} \sum_{t>0} \mathbf{M}(\phi, t)}{\sum_{\phi} \sum_{t<0} \mathbf{M}(\phi, t)} \approx 1 \tag{9}$$

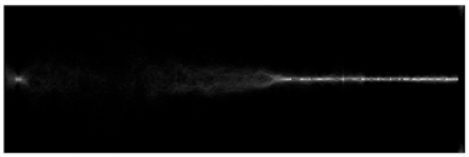
Figures 5(a) and 6(c) show the temporal slices of two shots which consist of different motions over time, while Figs. 5(b) and 6(d) show the corresponding tensor histograms. In Fig. 5, the motion is segmented into two sub-units, while in Fig. 6, the motion is segmented into three sub-units.

3.4. Background Segmentation

Figure 7 illustrates the major flow of our approach. Given a set of spatio-temporal slices,[7] a $2D$ tensor histogram is computed. The $2D$ histogram is further

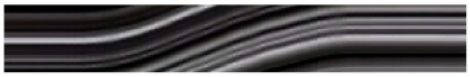


(a)

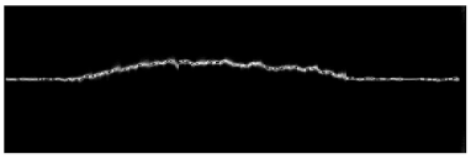


(b)

Figure 5. Zoom followed by static motion. (a) temporal slice; and (b) censor histogram.



(a)



(b)

Figure 6. Static, pan, and static motions. (a) temporal slice; and (b) tensor histogram.

non-uniformly quantized into a $1D$ normalized motion histogram. The histogram consists of seven bins to qualitatively represent the rigid camera and object motions. The peak of the histogram is back projected onto the original image sequence. The projected pixels are aligned and pasted to generate a complete background. With the background information, foreground objects can also be obtained through the background subtraction technique (Ngo, 2000).

3.4.1. Quantization of Motion Histogram. Given a $2D$ tensor histogram $\mathbf{M}(\phi, t)$ with temporally coherent motion unit, the tensor orientation ϕ is non-uniformly quantized into seven bins, where

$$\begin{aligned} \Phi_1 &= [-90^\circ, -45^\circ) & \Phi_5 &= (5^\circ, 25^\circ] \\ \Phi_2 &= [-45^\circ, -25^\circ) & \Phi_6 &= (25^\circ, 45^\circ] \\ \Phi_3 &= [-25^\circ, -5^\circ) & \Phi_7 &= (45^\circ, 90^\circ] \\ \Phi_4 &= (-5^\circ, 5^\circ] \end{aligned}$$

The scheme quantifies motion based on its intensity and direction. Φ_1 and Φ_7 represent the most intense motion, while Φ_4 represents no or slight motion. The normalized $1D$ motion histogram $\mathbf{N}$ is computed by

$$\mathbf{N}(\Phi_k) = \frac{\sum_{\phi_i \in \Phi_k} \sum_t \mathbf{M}(\phi_i, t)}{\sum_{k=1}^{7} \mathbf{N}(\Phi_k)} \tag{10}$$

Adaptive setting of quantization scale is a difficult problem. Since we assume motion characterization is performed prior to motion segmentation, camera motion is supposed to be coherent and smooth. Thus, the setting should not be too sensitive to the final results. Empirical results indicate that our proposed setting is appropriate for most cases.

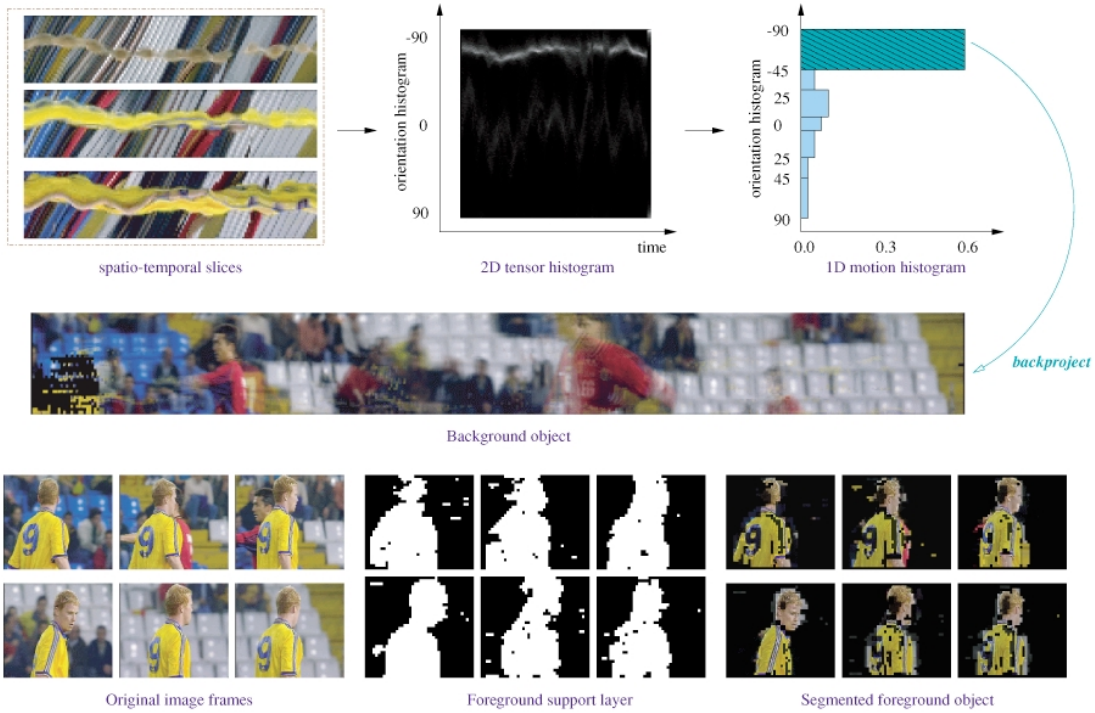


Figure 7. The original scheme for background segmentation.

3.4.2. Tensor Back-Projection. The prominent peak in a $1D$ motion histogram reflects the dominant motion of a sequence, as shown in Fig. 7. By projecting the peak back to the temporal slices $\mathbf{H}_i$, we can locate the region (referred to as the layer of support) that induces the dominant motion. The support layer is computed as,

$$\text{Mask}_i(x,t) = \begin{cases} 1 & \phi \in \hat{\Phi} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where

$$\hat{\Phi} = \arg\left\{ \max_{\Phi_k} \mathbf{N}(\Phi_k) \right\} \tag{12}$$

(x,t) is the location of a pixel in $\mathbf{H}_i$. Figure 8 illustrates an example: the temporal slice consists of two motions, while the layer of support locates the region corresponding to the dominant motion (white color). The result of localization is correct, except at the border of two motion patterns due to the effect of Gaussian smoothing prior to tensor computation.

3.4.3. Point Correspondence and Background Mosaicking. Once the support layer of a dominant motion is computed, in principle we can align and paste the corresponding regions to reconstruct the background image. Nevertheless, this is not a trivial issue since theoretically the correspondence feature points need to be matched across frames. This is an ill-posed problem specifically at the textureless regions. The problem is further complicated by occluded and uncovered feature points at a particular time instant.

To solve this problem, we propose a method that selects temporal slice $\mathbf{H}_i$ which contains two adjacent scans $\mathbf{H}_i(x,t)$ and $\mathbf{H}_i(x,t+1)$ with the most textural information at time t, and then perform feature points matching across the two scans. For each time instance t, the criterion for selecting a slice is

$$\hat{\mathbf{H}} = \arg\max_{\mathbf{H}_i} \left\{ \frac{\mathbf{C}_i(t) + \mathbf{C}_i(t+1)}{|n_i(t) - n_i(t+1)| + 1} \right\} \tag{13}$$

and

$$\mathbf{C}_i(t) = \sum_x c_i(x,t)\text{Mask}_i(x,t)$$

$$n_i(t) = \sum_x \text{Mask}_i(x,t)$$

where $c_i(x,t)$ is the certainty measure of a tensor at $\mathbf{H}_i(x,t)$ (see Eq. (5)). The value c_i indicates the richness of texture of pixels surrounding the pixel located at (x,t). In practice, $\mathbf{C}_i(t) > 0$ and $n_i(t) \geq 2$.

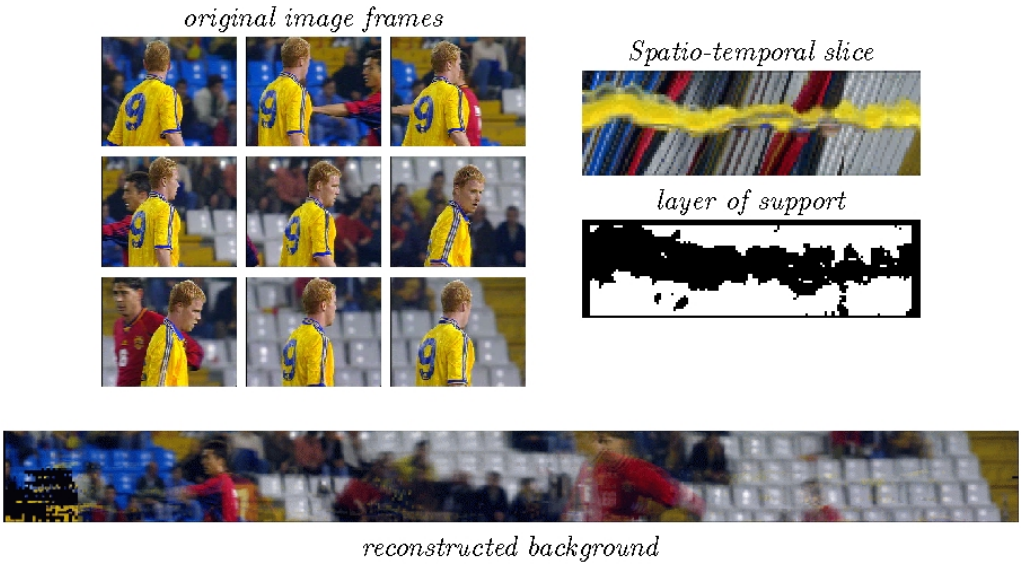


Figure 8. Background reconstruction.

For simplicity, we assume the motion model involves only translation when aligning and pasting two image frames. Let us denote $\hat{d}(t)$ as the translation vector at time t, $\hat{d}(t)$ is directly computed from two scans by

$$\hat{d}(t_1) = \arg\min_{d} \{\text{med} \mid \mathbf{H}_i(x, t) - \mathbf{H}_i(x + d, t + 1)|\} \tag{14}$$

where med is a robust median estimator employed to reject outliers. The value of d is set to $1 \leq d \leq 5$ while the sign of d is dependent on the $\hat{\Phi}_k$ in (12) which indicates the motion direction.[8] From (14), it is interesting to note that the problem of occlusion and uncovered regions is implicitly solved due to the use of support layer and robust estimator. Naturally the occluded region at frame i can be filled by the uncovered region at frame $j \neq i$. An example of a mosaicked background reconstructed from 140 frames is shown in Fig. 8.

4. Video Representation

A concrete way of describing video content for scene change detection is to represent each shot with background images. Nevertheless, such task is always non-trivial. Suppose no domain specific knowledge is utilized, it can only be achieved to a certain extent if more than one motion layer can be formed by camera and object movements. For instance, when a camera tracks a moving object, two layers are formed, one corresponds to the background and the other corresponds to the targeted object. In this case, the background object can be extracted and reconstructed as a panoramic image. However, if a camera just pans across a background and overlooks objects that do not move, the objects will be absorbed as part of the background image and only one motion layer will be formed.

Based on the current state-of-art in image sequence analysis, we propose a video representation strategy as illustrated in Fig. 9. The strategy consists of two major parts: keyframe selection and formation, and background reconstruction. The idea is to represent shots compactly and adaptively through motion characterization, at the same time, extract background objects as far as possible through motion segmentation. Because foreground objects will not be separated from background image for the single motion case, we will further discuss a method based on similarity measure in the next section to reduce the distraction of foreground objects when comparing background images.

4.1. *Keyframe Selection and Keyframe Formation*

Keyframe selection is the process of picking up frames directly from sub-units to represent the content of a

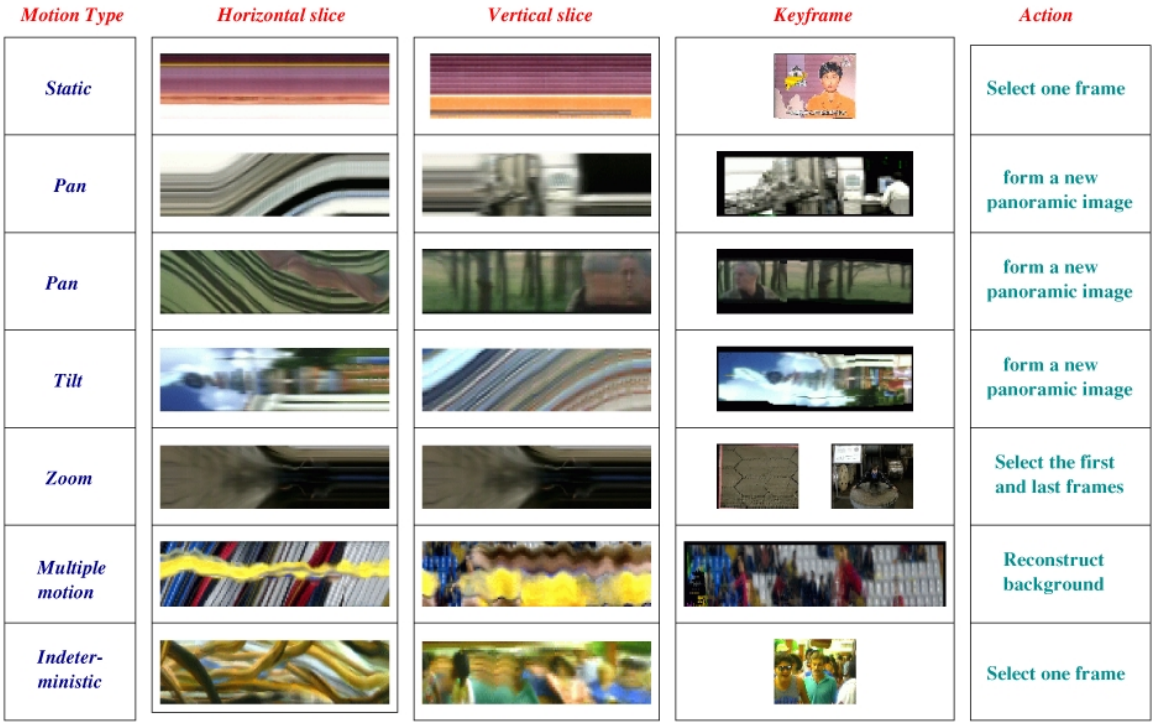


Figure 9. Keyframe selection and formation.

shot. On the other hand, keyframe formation is the process of forming a new image given a sub-unit. Whether to select or form images is directly related to the camera motion in a shot. For instance, a sub-unit with camera panning is well summarized if a new image can be formed to describe the panoramic view of the scene, instead of selecting few frames from the sequence. On the other hand, the content of a sub-unit with camera zooming is well summarized by just selecting two frames before and after zoom, instead of selecting few frames from the sequence. In our approach, with reference to Fig. 9, one frame is arbitrarily selected to summarize the content of a sub-unit with static or indeterministic motion, a panoramic image is formed for a sub-unit with camera panning or tilting, and two frames are selected for a sub-unit with camera zoom. For indeterministic motion, a sub-unit normally lasts for less than ten frames, hence, one frame is generally good enough to summarize the content.

4.2. *Background Reconstruction*

Scene is normally composed of shots that are shot at the same place. Intuitively, background objects are more important than foreground objects in grouping similar shots as a scene. Given an image sequence with both camera and object motions, our aim is to reconstruct a background scene after segmenting the background and foreground layers. We assume here the dominant motion layer always corresponds to the background layer. The background is reconstructed based on the techniques described in Section 3.3. Each background image is associated with a support layer for similarity measure.

5. Similarity Measure

Let the representative frames of shot s_i be $\{r_{i1}, r_{i2}, \ldots, r_{ik}\}$. The similarity between the two shots s_i and s_j is defined as

$$\mathrm{Sim}(s_i, s_j) = \frac{1}{2}\{\mathcal{M}(s_i, s_j) + \hat{\mathcal{M}}(s_i, s_j)\} \quad (15)$$

where

$$\mathcal{M}(s_i, s_j) = \max_{p=\{1,2,\ldots\}} \max_{q=\{1,2,\ldots\}} \{\mathrm{Intersect}(r_{ip}, r_{jq})\} \quad (16)$$

$$\hat{\mathcal{M}}(s_i, s_j) = \hat{\max}_{p=\{1,2,\ldots\}} \max_{q=\{1,2,\ldots\}} \{\mathrm{Intersect}(r_{ip}, r_{jq})\} \quad (17)$$

mâx is the second largest value among all pair of keyframe comparisons. The disadvantage of using color histograms as features is that two keyframes will be considered similar as long as they have similar color distributions, even through their contents are different. To remedy this deficiency, we use not only $\mathcal{M}$ but also $\hat{\mathcal{M}}$ for the sake of robustness.

The color histogram intersection, Intersect(r_i, r_j), of two frames r_i and r_j is defined as

$$\text{Intersect}(r_i, r_j) = \frac{1}{\mathcal{A}(r_i, r_j)} \times \sum_h \sum_s \sum_v \times \min\{H_i(h, s, v), H_j(h, s, v)\} \quad (18)$$

where

$$\mathcal{A}(r_i, r_j) = \min\left\{ \sum_h \sum_s \sum_v H_i(h, s, v), \sum_h \sum_s \sum_v H_j(h, s, v) \right\} \quad (19)$$

$H_i(h, s, v)$ is a histogram in HSV (hue, saturation, intensity) color space. Because hue conveys the most significant characteristic of color, it is quantized to 18 bins. Saturation and intensity are each quantized into 3 bins. This quantization provides 162 ($18 \times 3 \times 3$) distinct color sets.

In (18), the degree of similarity is proportional to the region of intersection. Intersect(r_i, r_j) is normalized by $\mathcal{A}(r_i, r_j)$ to obtain a fractional similarity value between 0 and 1. For instance, given an image frame $\mathbf{I}$ of size $m \times n$ and a background image **Bg** of size $M \times N$ ($m < M, n < N$), Eq. (18) gives the fractional region in $\mathbf{I}$ which overlaps with **Bg** (see Fig. 10 for illustration). Color Histogram intersection can reduce the effect of:

- the distraction of foreground objects[9]
- viewing a site from a variety of viewpoints
- occlusion
- varying image resolution

The last three items are consequences of employing color histograms as image features, while the first item is due to the use of the histogram intersection. Figure 10 illustrates an example. The similarity of $\mathbf{I}_i$ and **Bg** (*Intersect*$_1$), and the similarity of $\mathbf{I}_j$ and **Bg** (*Intersect*$_2$) directly correspond to their overlapping area of background. In contrast to the Euclidean distance measure, which takes a foreground object into consideration, the histogram intersection, intuitively, is a more suitable similarity measure for scene change detection. Nevertheless, it should be noted that the intersection of $\mathbf{I}_i$ and $\mathbf{I}_j$ corresponds to the foreground player. Here, segmentation which is a difficult task, needed to be done prior to the similarity measure!

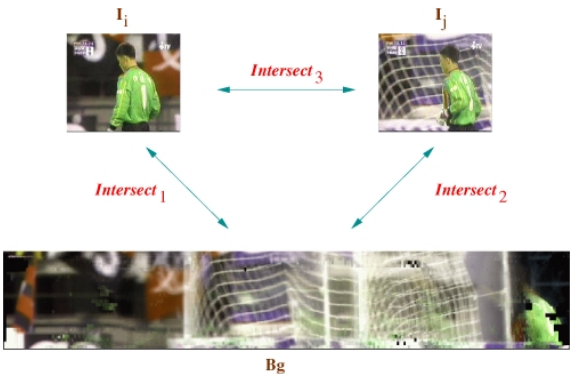


Figure 10. Histogram intersection. *Intersect*$_1(\mathbf{I}_i, \mathbf{Bg})$ and *Intersect*$_2(\mathbf{I}_j, \mathbf{Bg})$ correspond to the background object, while *Intersect*$_3(\mathbf{I}_i, \mathbf{I}_j)$ correspond to the foreground player.

To detect scene changes, we need a similarity threshold T_s to decide if two shots belong to a same scene. Threshold setting is a common practice but tedious experience for most computer vision tasks. Here, we describe a method to adaptively set thresholds by taking into account the characteristics of videos. Denote n as the number of shots in a video, the threshold T_s of a video is defined as

$$T_s = \mu + \sigma \quad (20)$$

where

$$\mu = \frac{2}{n \times (n-1)} \left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{Sim}(s_i, s_j) \right\} \quad (21)$$

$$\sigma = \frac{2}{n \times (n-1)} \sqrt{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \{\mu - \text{Sim}(s_i, s_j)\}^2} \quad (22)$$

μ and σ are respectively the mean and the standard deviation of the similarity measures among all pairs of shots.

6. Time-Constraint Grouping

The idea is that the probability of two shots belonging to the same scene is directly related to their tempo-

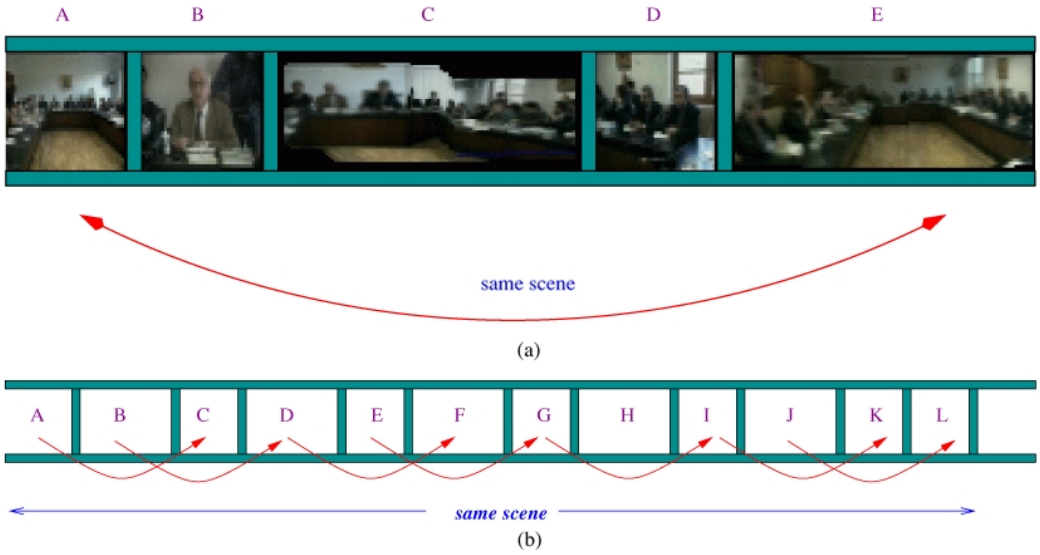


Figure 11. Time constraint grouping. (a) shots in one scene; and (b) red arrows indicate similar shots.

ral distance. In other words, two shots s_i and s_j will not be grouped in the same scene if they are temporally far apart. For simplicity, we consider only the case where a scene is composed of temporally contiguous shots. Using Fig. 11(a) as an example, suppose that shots A and E are determined to be similar by Eq. (20), we group all shots from A to E as one scene even shots B, C, D may be considered dissimilar to A and E.

The algorithm runs in the following way: at shot s_i, it looks forward at most c shots. If s_i and s_{i+c} are similar, then $\forall_{i\le j\le i+c} s_j$ are grouped in one scene. Notice that this algorithm will not limit the duration of a scene. As shown in Fig. 11(b), shots are grouped progressively in one scene until there is no similar shot found within the temporal distance c. Rigorously, a group of adjacent shots $\{s_m, s_{m+1}, \ldots, s_{n-1}, s_n\}$ is clustered in a scene if the following conditions are fulfilled

- *Condition 1*: $\exists t$ such that $t = \arg\{\max_{r=\{1,2,\ldots,c\}} \mathrm{Sim}(s_m, s_{m+r})\}$, $\mathrm{Sim}(s_m, s_{m+t}) \ge T_s$, and $\forall_{r=\{1,2,\ldots,c\}} \mathrm{Sim}(s_{m-r}, s_m) < T_s$.
- *Condition 2*: $\exists t$ such that $t = \arg\{\max_{r=\{1,2,\ldots,c\}} \mathrm{Sim}(s_{n-r}, s_n)\}$, $\mathrm{Sim}(s_{n-t}, s_n) \ge T_s$, and $\forall_{r=\{1,2,\ldots,c\}} \mathrm{Sim}(s_n, s_{n+r}) < T_s$.
- *Condition 3*: $\exists t_1, t_2$ such that $\{t_1, t_2\} = \arg\{\max_{r=\{0,1,2,\ldots,c\}, s=\{0,1,2,\ldots,c\}} \mathrm{Sim}(s_{i-r}, s_{i+s})\}$, $\mathrm{Sim}(s_{i-t_1}, s_{i+t_2}) \ge T_s$, $m < i < n$ and $0 < |t_1 - t_2| \le c$.

where $\mathrm{Sim}(s_i, s_j)$ is the similarity measure between the shots i and j and T_s is the similarity threshold. The parameter c is a constraint which is used as follows: suppose $j - i \le c$, $i < j$ and $\mathrm{Sim}(s_i, s_j) \ge T_s$, then $\forall_{i\le k\le j}\ s_k$ are clustered in one scene.

Condition 1 states that the first shot of a scene must have at least one similar shot succeeding it within the distance c (shots A and C in Fig. 11(b)). Similarly, Condition 2 states that the last shot of a scene must have at least one similar shot preceding it within c (shots L and J in Fig. 11(b)). Condition 3 states that s_i, $m < i < n$, is either similar to a shot preceding (shots G and E in Fig. 11(b)) or succeeding s_i (shots B and D in Fig. 11(b)), or at least one shot preceding s_i is similar to a shot succeeding s_i within c (shot H in Fig. 11(b)).

In the experiments, the parameter c is set to a value such that $f_{i+c} - f_i \le 900 < f_{i+c+1} - f_i$, where f_i is the start time (in frame unit) of a shot s_i. In other words, at shot s_i, shot s_{i+c} that is less than or equal to 900 frames (about 30 second) apart from s_i is compared for similarity.

7. Experiments

Figure 12 shows an example for the detailed procedure of the proposed scene change detection framework on a news video *demo.mpg*. For simplicity, we only show the horizontal spatio-temporal slice extracted from the news video. This slice is first partitioned into twelve

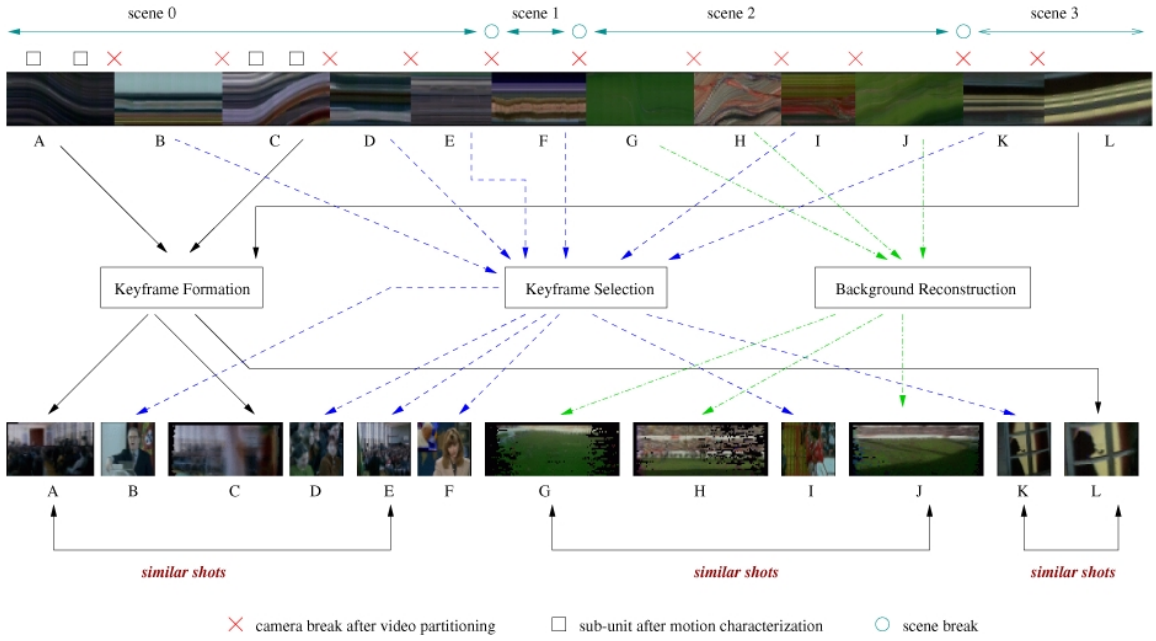


Figure 12. An illustration for the scene change detection framework tested on the video *demo.mpg*.

shots using the video partitioning algorithm, and then the tensor histogram is computed for each shot. These shots are further temporally segmented into finer sub-units and are annotated based on the proposed motion characterization method. As shown in the figure, shots A and C are segmented into sub-units with static and panning motions. Based on the annotated motions, keyframes are adaptively selected (shots B, D, E, F, I and K) and formed (shots A, C, L), in addition, backgrounds are reconstructed (shots G, H, J) for multiple motion cases. Finally, color features are extracted from each keyframe for similarity measure through histogram intersection. As indicated in the figure, shots A and E (G and J) are considered similar, as a results, all shots from A to E (G and J) are grouped as one scene based on the time-constraint grouping algorithm.

We conducted experiments on other four videos[10]: *father.mpg*, *Italy.mpg*, *lgerca_lisa_1.mpg* and *lgerca_lisa_2.mpg*. Table 1 shows the experimental results on the videos *father.mpg* and *Italy.mpg*. Both videos have indoor and outdoor scenes. Initially, shots that happened at the same sites are manually grouped as scenes and served as ground truth data. The data is then compared with the results generated by our approach. Experimental results show that the proposed approach works reasonably well in detecting most of the scene boundaries (e.g., boundaries between indoor-outdoor scenes, indoor-indoor scenes and outdoor-outdoor scenes). The only false detection in *Italy.mpg* is due to the significant change of background color

Table 1. Experimental results.

Scene	Shots	C	F	M
	father.mpg			
0	0-0	1	0	0
1	1-1	1	0	0
2	2-2	1	0	0
3	3-3	1	0	0
4	4-8	1	0	0
5	9-9	1	0	0
6	10-14	1	0	0
7	15-16	1	0	0
8	17-23	1	0	0
	Italy.mpg			
0	0-2	1	1	0
1	3-3	1	0	0
2	4-4	1	0	0
3	5-13	1	0	0
4	14-19	1	0	0
5	20-38	0	0	1

C: Correct detection, F: False detection, M: Missed detection.

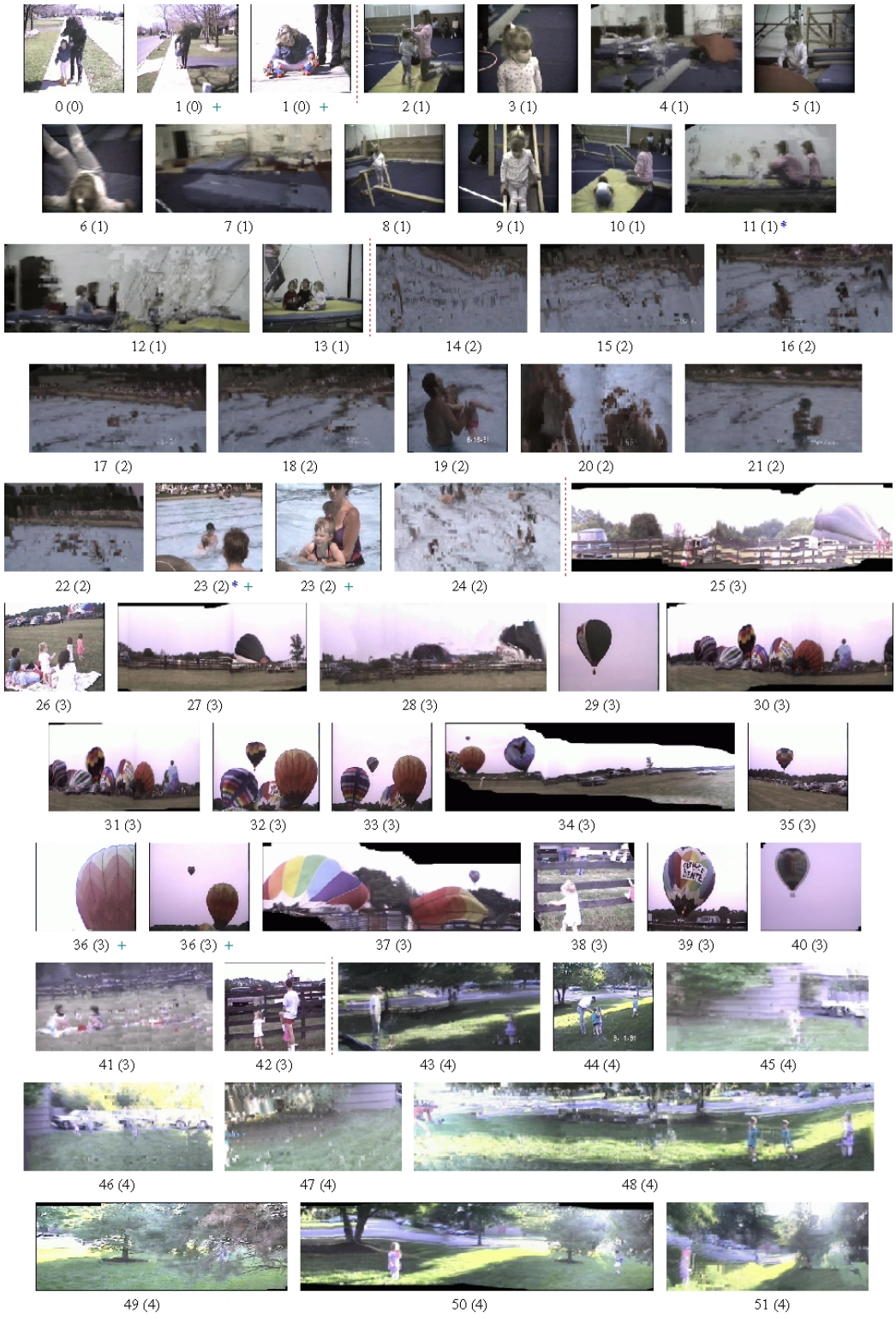


Figure 13. Some keyframes of *lgerca_lisa_1.mpg*. X(Y): shot(scene). (* Indicates false alarm, + indicates zoom, and dotted vertical bars indicate scene boundaries.)

Figure 14. Some keyframes of *lgerca_lisa_2.mpg*. X(Y): shot(scene). (* Indicates false alarm, + indicates zoom, and dotted vertical bars indicate scene boundaries.)

Table 2. Experimental results on *lgerca_lisa_1.mpg*.

Scene	Scene description	Shots	C	F	M
0	Kids learning roller-skater	0-1	1	0	0
1	Kids playing in gym	2-13	1	1	0
2	Kids playing with water with parent	14-24	1	1	0
3	Hot balloon event	25-42	1	0	0
4	Kids playing on lawn	43-51	1	0	0

C: Correct detection, F: False detection, M: Missed detection.

Table 3. Experimental results on *lgerca_lisa_2.mpg*.

Scene	Scene description	Shots	C	F	M
0	Kid at home with cat	0-1	1	0	0
1	Kids in gym	2-8	1	0	0
2	Kids playing high-bar	9-12	1	1	0
3	Kids + teacher with high-bar	13-14	1	0	0
4	Kids jumping	15-15	1	0	0
5	Kids in gym	16-17	1	0	0
6	Kids in gym (over-illuminated)	18-28	1	0	0
7	Kids playing at home	29-31	1	0	0
8	Kid driving outside home	32-36	1	0	0
9	Kids dancing (I)	37-39	1	0	0
10	Kids dancing (II)	40-40	1	0	0
11	Kids dancing (III)	41-42	0	0	1
12	After play	43-51	1	0	0
13	Swimming pool	52-53	0	0	1
14	Crowded in swimming pool	54-55	0	0	1

C: Correct detection, F: False detection, M: Missed detection.

in an indoor scene, while the only missed detection in *Italy.mpg* is due to the similar background color distribution between an indoor scene and an outdoor scene.

Tables 2 and 3 show the experimental results on the two MPEG-7 test videos, *lgerca_lisa_1.mpg* and *lgerca_lisa_2.mpg*. Both are home videos and each video has approximately 32,000 frames. The experimental results are compared with the ground truth data provided by MPEG-7 test sets. In *lgeraca_lisa_1.mpg*, the two false alarms are due to illumination effect. In *lgeraca_lisa_2.mpg*, the results of the two missing scenes are arguable since these scenes are composed of shots in the same places (scenes 10-11 have taken place on stage, scenes 13-14 have taken place in a swimming pool). Figures 13 and 14 show some of keyframes in the two tested videos.

Table 4. Speed efficiency (on a Plentium-III platform).

Step	*lgerca_lisa_1.mpg*	*lgerca_lisa_2.mpg*
Video partitioning	800 sec	805 sec
Motion characterization to keyframe generation	4080 sec (1.14 hour)	3840 sec (1.06 hour)
Similarity measure and time constraint grouping	103 sec	105 sec

Table 4 shows the speed efficiency of the proposed scene change detection framework on the two tested videos. For video partitioning, our approach operates in real time, approximately 40 frames per second on a Plentium-III machine. As indicated in the table, the procedure from motion characterization to keyframe generation (including time to generate color feature vector for each keyframe) consumes most of the processing time. For similarity measure, most of the processing time is spent on finding the adaptive threshold in Eq. (20).

8. Conclusion

A motion-based video representation scheme has been proposed for scene change detection by integrating the motion characterization and background reconstruction techniques. Using this scheme, an adaptive keyframe selection and formation method has been derived. By combining the histogram intersection for similarity measure and the time constraint grouping algorithm, encouraging experimental results have been reported. We expect that the results can be further improved if background segmentation and reconstruction can be done for shots either with static or non-static motion prior to measuring shot similarity.

Acknowledgments

This work is supported in part by RGC Grants HKUST661/95E and HKUST6072/97E.

Notes

1. For instance, Rui et al. (1998) also takes shot activity measure into consideration.
2. Slanted lines in horizontal slices depict camera panning, while slanted lines in vertical slices depict camera tilting.
3. DC image is formed by using the first coefficient of each 8×8 Discrete Cosine Transform (DCT) block.

4. The algorithm introduced by Yeo and Liu (1995) is applied to estimate DC components from P-frames and B-frames.
5. To supress noise, each slice is smoothed by a 3×3 Gaussian kernel prior to tensor computation.
6. The tensor histograms of both horizontal and vertical slices are utilized. A sequence is characterized as zoom if either one of the histograms satisfies (9).
7. Figure 7 only shows three horizontal spatio-temporal slices extracted from different rows of an DC image volume. They illustrate the motion patterns in the top (2nd row), middle (18th row) and bottom (34th row) portions of the image volume.
8. It is worthwhile to notice that $\hat{\Phi}_k$ can tell the range of d. However, this information is not exploited in (14) since the computational save in predicting d by $\hat{\Phi}_k$ is insignificant. This is due to the fact that only small amount of data (two columns of pixels) is used to compute (14).
9. This feature is useful when different foreground objects appear in a background image at different time instant.
10. The first two videos can be obtained from *http://mmlib.cs.ust.hk/scene.html*, the last two videos are MPEG-7 standard test videos.

References

Corridoni, J.M. and Del. Bimbo, A. 1998. Structured representation and automatic indexing of movie information content. *Pattern Recognition*, 31(12):2027–2045.

Flickner M. et al. 1995. Query by image and video content: The QBIC system. *Computer*, 28(9):23–32.

Gudivada, V.N. and Raghavan, V.V. 1995. Introduction: Content-based image retrieval systems. *Computer*, 28(9):18–22.

Hanjalic, A., Lagendijk, R.L., and Biemond, J. 1999. Automated high-level movie segmentation for advanced video retrieval systems. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(5):580–588.

Huang, J., Liu, Z., and Wang, Y. 1998. Integration of audio and visual information for content-based video segmentation. *Intl. Conf. on Image Processing.*, vol. 3, pp. 526–529.

Jacobs, D.W., Weinshall, D., and Gdalyahu, Y. 2000. Classification with non-metric distances: Image retrieval and class representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(6):583–600.

Jähne, B. 1991. *Spatio-Temporal Image Processing: Theory and Scientific Applications*. Springer Verlag: Berlin.

Ngo, C.W. 2000. Analysis of spatio-temporal slices for video content representation. Ph.D. Thesis, Hong Kong University of Science and Technology.

Ngo, C.W., Pong, T.C., and Chin, R.T. 1999. Detection of gradual transitions through temporal slice analysis. *Computer Vision and Pattern Recognition*, 1:36–41.

Ngo, C.W., Pong, T.C., and Chin, R.T. 2000a. A robust wipe detection algorithm. In *Asian Conf. on Computer Vision*, vol. 1, pp. 246–251.

Ngo, C.W., Pong, T.C., and Chin, R.T. 2001. Video partitioning by temporal slice coherency. *IEEE Trans. on Circuits and Systems for Video Technology*.

Ngo, C.W., Pong, T.C., Zhang, H.J., and Chin, R.T. 2000b. Motion characterization by temporal slice analysis. *Computer Vision and Pattern Recognition*, 2:768–773.

Rui, Y., Huang, T.S., and Mehrotra, S. 1998. Exploring video structure beyong the shots. In *Proc. IEEE Conf. on Multimedia Computing and Systems*, pp. 237–240.

Sahouria, E. and Zakhor, A. 1999. Content analysis of video using principle components. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(8):1290–1298.

Santini, S. and Jain, R. 1999. Similarity matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*.

Sundaram, H. and Chang, S.F. 2000. Determining computable scenes in films and their structure using audio-visual memory models. *ACM Multimedia*.

Swain, M.J. and Ballard, D.H. 1991. Color indexing. *Int. Journal of Computer Vision*, 7(1):11–32.

Yeo, B.L. and Liu, B. 1995. On the extraction of DC sequence from MPEG compressed video. *IEEE Int. Conf. on Image Processing*, 2:260–263.

Yeung, M.M. and Yeo, B.L. 1997. Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Trans. on Circuits and Systems for Video Technology*, 7(5):771–785.