6-1999

Camera break detection by partitioning of 2D spatio-temporal images in MPEG domain

Chong-Wah NGO

Citation

NGO, Chong-Wah. Camera break detection by partitioning of 2D spatio-temporal images in MPEG domain. (1999). *Proceedings of the 1999 International Conference on Multimedia Computing and Systems, Florence, Italy, June 7-11*. 1, 750-755.

Available at: https://ink.library.smu.edu.sg/sis_research/6612

Camera Break Detection by Partitioning of 2D Spatio-temporal Images in MPEG Domain

C. W. Ngo, T. C. Pong & R. T. Chin
Department of Computer Science
The Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong
Email:{cwngo, tcpong, roland) @cs.ust.hk

Abstract

In this paper, we propose a new approach to detect camera cuts and wipes. The approach projects a video into two images representing the spatio-temporal continuity of a video life. Visually, these images are composed by regions of different patterns, each region represents the temporal location of a shot. The shape of a region boundary inherently classifies cut and wipe, and most importantly, marks the start and end of a wipe sequence. We hence formulate algorithms to locate the color and texture discontinuities occurring at the boundaries of image regions.

Keywords spatio-temporal image segmentation, camera cut and wipe detection.

1 Introduction

A video is physically formed by shots, where each shot is one or more frames in series on a continuous length of film stock [2]. We perceive a shot as an uninterrupted segment of screen time, space and graphical configurations. The boundary between two shots is referred to as a camera break. We call a camera break as cut if there is an instantaneous change from one shot to another; a camera break as gradual transition (e.g., wipe and dissolve) if one shot gradually interrupts and replaces another shot. For instance, in a wipe, one frame replaces another as a moving boundary line crosses the screen. By decomposing videos into shots, we can reduce video search problems to image search problems, in addition, facilitate the nonlinear browsing of video content.

In the current literature, there are various algorithms [7, 11, 8, 5] for detecting camera breaks, in general, we can categorize them as statistic-based, histogram-based, feature-based, transform-based, and motion-based. These methods, in general, are based on the frame-to-frame similarity measure. Although spatial and temporal sub-sampling of video frames are suggested to improve speed efficiency [10], the success still depends on the choice of the spatial window size and the temporal sampling step. Smaller window size is sensitive to object and camera motions while larger sampling step can easily skip fragmented shots.

In this paper, we propose a new approach on detecting camera breaks directly in the MPEG domain. Similar to [4], the approach is based on the analysis of two spatio-temporal images projected from an image sequence, except that [4] only addressed camera cut detection in the uncompressed domain. The projected images are composed of spatially and temporally coherent regions, indicating the temporal location of cuts and wipes. In principle, segmenting these images into regions is equivalent to detecting camera breaks. Compared with other techniques, this approach can detect as well as classify cut and wipe, in addition, offers the advantage of detecting the exact duration of a wipe. The cut detection, nevertheless, is sensitive to fast object motion at the center of a frame. To eliminate false alarms, further investigation of the MPEG motion vectors and DCT coefficients of those frames located at the boundaries of suspected shots are required.

2 Spatio-Temporal Image Model

2.1 Video Slices

Denote f_{dc} as a M x N DC image[1], f_{dc} is vertically and horizontally projected to two $1D$ slices v and h,

$$v(i) = \sum_{p=k1-j}^{k1+j} \alpha_p f_{dc}(p,i), \text{ where } k1 = \frac{M}{2} \quad (1)$$

$$h(i) = \sum_{p=k2-j}^{k2+j} \alpha_p f_{dc}(i,p), \text{ where } k2 = \frac{N}{2} \quad (2)$$

where $0 \leq p < M$ or N, and $\sum \alpha_p = 1$. When $j = 0$, the middle row and column of f_{dc} are taken to form

[1] The DC images of P-frames and B-frames are estimated using method proposed by [9]

0-7695-0253-9/99 $10.00 © 1999 IEEE

the slices. To ensure the smoothness of vertical and horizontal slices within a shot, we set j = 1 and perform Gaussian smoothing on the slices. The Gaussian kernel used is $\alpha = [0.2236, 0.5477, 0.2336]$. By cascading these slices over time, this model acquires a 2D image $\mathbf{V}$ formed by vertical slices and a 2D image H formed by horizontal slices. Denote t as the time coordinate and $(x, \mathbf{y})$ as the image coordinate, then H and $\mathbf{V}$ are in x $- t$ and y $- t$ space respectively. H and $\mathbf{V}$ are spatio-temporal images modeling motions in two orthogonal directions. The projection is similar to slicing video frames along the temporal direction.

Figure 1 shows the spatio-temporal images of a news video sequence connected by three shots; Figure 2 shows the spatio-temporal images of two shots connected by a wipe. As seen in the figure, each image contains several spatially uniform texture regions, and in fact, each region is formed by the slices taken from frames that belong to the same shot. Notice that the type of camera breaks will affect the boundary shape of two connected regions. A camera cut results in a vertical boundary line in both $\mathbf{V}$ and $\mathbf{H}$ (see Figure 1); while a wipe results in a slanted boundary line in one of the spatial-temporal images (see Figure 2). Figure 3 further illustrates the patterns of image regions as a result of various wipe directions.

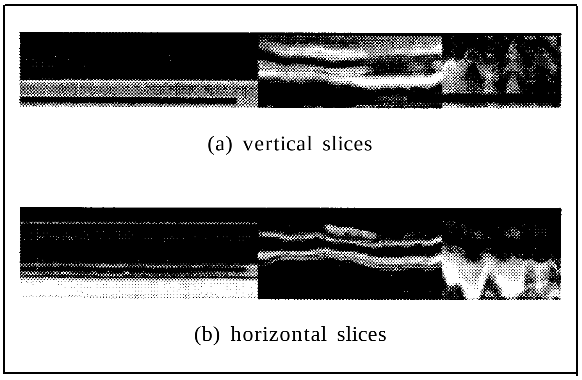
Based on these model characteristics, we claim that the task of detecting camera breaks is equivalent to the task of detecting region boundaries. Furthermore, by investigating the two end points and the orientation of a boundary, this model offers not only the capability of classifying cut and wipe, but also knowledge on the duration of a wipe sequence. In contrast to the wipe detector [1] based on the statistical measure proposed by Alattar, this model, in principle, can tolerate vigorous motion without causing false alarms. In the following subsections, we propose a Markov based image segmentation algorithm to locate the color-texture discontinuities at region boundaries.

2.2 Computing Color-Texture Features

Denote $\mathbf{H} = [H_r, H_g, H_b, H_y]$ and $\mathbf{V} = [V_r, V_g, V_b, V_y]$ as the spatio-temporal images in (r, g, b) color space[2] and y luminance space. The approach computes edge information by

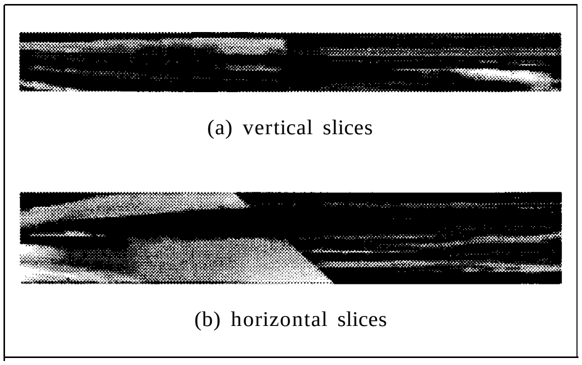
$$E^{H_i}_{\sigma,\theta} = \bar{\mathbf{GD}}_{\sigma,\theta} * H_i \qquad (3)$$

where $*$ is a convolution operator and $i \in \{r, g, \text{b}\}$. $\bar{\mathbf{GD}}_{\sigma,\theta}$ is the first derivative Gaussian along the x-axis


(a) vertical slices

(b) horizontal slices

Figure 1: Spatio-temporal images: (a) vertical (b) horizontal slices of a news sequence connected by three shots.


(a) vertical slices

(b) horizontal slices

Figure 2: Spatio-temporal images: (a) vertical (b) horizontal slices of two shots connected by a wipe.

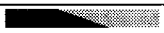
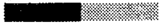
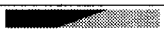
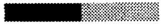
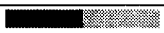
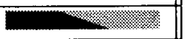
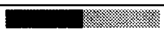
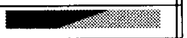
wiping direction	horizontal slices	vertical slices
left-to-right		
right-to-left		
top-to-bottom		
bottom-to-top		

Figure 3: The image patterns created by various wipe sequences.

“Note that MPEG uses YCrCb color space; our method converts the YCrCb to RGB components given by

$$\bar{\mathbf{GD}}_{\sigma,\theta}(x, \mathrm{Y}) = -\frac{x}{\sigma^2}\bar{\mathbf{G}}_{\sigma,\theta}(x, \mathrm{Y}) \tag{4}$$
$$\bar{\mathbf{G}}_{\sigma,\theta}(x,y) = \bar{\mathbf{G}}_{\sigma}(x', y')$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$; $\bar{\mathbf{G}}_{\sigma}(x, \mathrm{Y}) = \exp\{-\frac{x^2+y^2}{2\sigma^2}\}$ is a Gaussian filter controlled by a smoothing parameter σ.

The texture feature is computed based on Gabor decomposition [3]. The complex Gabor images are,

$$T_{\sigma_x,\sigma_y,\theta} = \hat{\mathbf{G}}_{\sigma_\mathbf{x},\sigma_\mathbf{y},\theta} * H_y \tag{5}$$

The Gabor filter $\hat{\mathbf{G}}_{\sigma_\mathbf{x},\sigma_\mathbf{y},\theta}(x,y) = \hat{\mathbf{G}}_{\sigma_\mathbf{x},\sigma_\mathbf{y}}(x', y')$ is expressed as

$$\hat{\mathbf{G}}_{\sigma_\mathbf{x},\sigma_\mathbf{y}}(x,y) \tag{6}$$
$$= (\frac{1}{2\pi\sigma_x\sigma_{\mathbf{Y}'}})\exp\{-\frac{1}{2}(\frac{x^2}{\sigma_2^2} + \frac{y^2}{\sigma_y^2})\}\exp\{2\pi jWx\}$$

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

Since a wipe normally lasts for one to two seconds (about 45 frames), we empirically set $\theta = \{0^o, 45^o, 135^o\}$. In addition, we set u = v = 0.4 and fix the values of σ, σ_x and σ_y, as a result, the color-texture feature is a twelve dimensional feature vector.

2.3 Segmenting Spatio-Temporal Image

The probability that a pixel pair $\eta = (\eta_h, \eta_v)$ at $\mathbf{H}(k,t)$ and $\mathbf{V}(k,t)$ is on the region boundary ξ of two connected regions is

$$p(\eta \in \xi | \mathbf{H}, \mathbf{V}) = p(\eta \in \xi | \mathbf{H}_N, \mathbf{V}_N) \tag{7}$$

where $\mathbf{H}_N$ and $\mathbf{V}_N$ are a 3 x3 neighborhood system shown in Figure 4. Based on the neighborhood system, we define eight connected components $C = \{C_1, C_2, \ldots, C_8\}$ (see Figure 5) to characterize η.

V1	V2	V3
V4	x	V6
V7	V8V8	V9

Figure 4: **The neighborhood system of a pixel x in** VN (the figure is also applicable to $\mathbf{H}_N$).

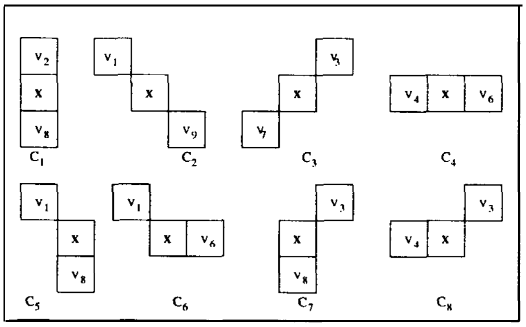


Figure 5: **The connected components defined on the neighborhood system.**

Assuming $\mathbf{H}_N$ and $\mathbf{V}_N$ are independent, we rewrite (7) as,

$$p(\eta | \mathbf{H}, \mathbf{V}) = p(\eta_h | \mathbf{H}_N) p(\eta_v | \mathbf{V}_N) \tag{8}$$

and $p(\eta_h), p(\eta_v)$ are the probability of η_h and η_v on a region boundary of H and V respectively. By Markov-Gibbs equivalence, we have

$$p(\eta_i) = \frac{1}{Z}\exp\{-U(\eta_i)\} \tag{9}$$

where Z is a normalizing constant, $i \in \{h, v\}$, and $U(\eta_i)$ is a energy function. The energy

$$U(\eta_i) = \sum_{c \in C} \beta_c \Gamma_c(\eta_i) \tag{10}$$

is the weighted sum of potential energy $\Gamma_c(\eta_i)$ over all connected components, where $\sum_{c \in C} \beta_c = 1$.

For classification and segmentation purpose, we further define three types of energy: $U_{cut}(\eta_i)$,$U_{wipe-}(\eta_i)$, $U_{wipe+}(\eta_i)$. For simplicity, we focus on image H_r first. Let η_h^r as a pixel locates at the (k,t) of H_r image, we have

$$\begin{bmatrix} U^r_{cut}(\eta^r_h) \\ U^r_{wipe-}(\eta^r_h) \\ U^r_{wipe+}(\eta^r_h) \end{bmatrix} = 3\begin{bmatrix} \Gamma^r_{C_1}(\eta^r_h) \\ \Gamma^r_{c'}(\eta^r_h) \\ \Gamma^r_{c''}(\eta^r_h) \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$
$$\begin{bmatrix} \Gamma^r_{C_1}(\eta^r_h) \\ \Gamma^r_{C_2}(\eta^r_h) \\ \Gamma^r_{C_3}(\eta^r_h) \end{bmatrix} - \begin{bmatrix} \Gamma^r_{C_4}(\eta^r_h) \\ \Gamma^r_{C_4}(\eta^r_h) \\ \Gamma^r_{C_4}(\eta^r_h) \end{bmatrix} \tag{11}$$

where

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

U^r_{cut} will give low energy if η^r_h is located at the region boundary as a result of a camera cut. Similarly, U^r_{wipe-} and U^r_{wipe+} will give low energy if η^r_h is located at the region boundary as a result of a camera wipe. The values of U^r_{wipe-} and U^r_{wipe+} depend on whether a boundary has negative or positive gradient.

Let $\eta_1 = (k_{\eta_1}, t_{\eta_1})$ and $\eta_2 = (k_{\eta_2}, t_{\eta_2})$ be the neighbors of η^r_h such that $\{\eta_1, \eta^r_h, \eta_2\}$ forms a connected component C_i. The potential energy is

$$\begin{aligned}\Gamma^r_{C_i}(\eta^r_h) &= \sum_\theta \{|E^{H_r}_{\sigma,\theta}(k,t) - E^{H_r}_{\sigma,\theta}(k_{\eta_1}, t_{\eta_1})| + \\ &\quad |E^{H_r}_{\sigma,\theta}(k,t) - E^{H_r}_{\sigma,\theta}(k_{\eta_2}, t_{\eta_2})|\} \qquad (12)\end{aligned}$$

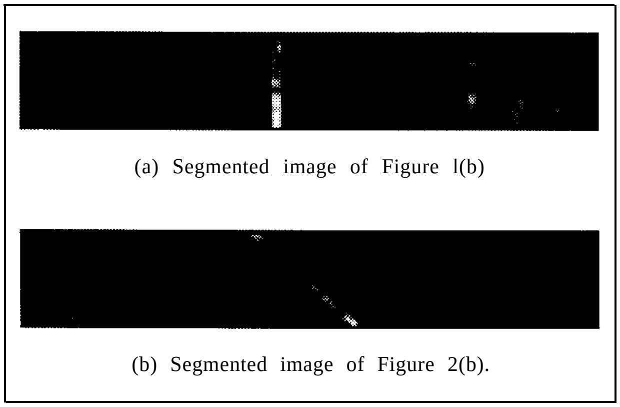
$\Gamma^g_{C_i}$ and $\Gamma^b_{C_i}$ are computed in a similar way; and $\Gamma^y_{C_i}$ is computed by

$$\begin{aligned}\Gamma^y_{C_i}(\eta^y_h) &= \sum_\theta \{|T_{\sigma_x,\sigma_y,\theta}(k,t) - T_{\sigma_x,\sigma_y,\theta}(k_{\eta_2}, t_{\eta_2})| + \\ &\quad |T_{\sigma_x,\sigma_y,\theta}(k,t) - T_{\sigma_x,\sigma_y,\theta}(k_{\eta_1}, t_{\eta_1})|\} \qquad (13)\end{aligned}$$

where η^y_h locates at the (k,t) of H_y and $\{\eta_1, \eta^y_h, \eta_2\}$ forms a connected component C_i. Subsequently, we define

$$U_{cut}(\eta_h) = \alpha_c \min_{j\in\{r,g,b\}} U^j_{cut}(\eta^j_h) + \alpha_t U^y_{cut}(\eta^y_h) \qquad (14)$$

where α_c and α_t are two parameters for weighting color and texture features. Similar approach is used to compute U_{wipe+} and U_{wipe-}. Figure 6 show the segmentation results with $\alpha_c = \alpha_t = 0.5$, the white lines which indicate the presence of low energy running across the boundaries of connected regions



(a) Segmented image of Figure l(b)

(b) Segmented image of Figure 2(b).

Figure 6: Segmentation result.

3 Shot Pruning

The approach introduced in the previous section can successfully detect cuts, however, false alarms may still happen due to the fast object motions at the center of frames and sharp illumination changes. In this section, we propose two approaches to prune false alarms. Since only the last frame L^{S_i} of a potential shot S_i are compared with the first frame $F^{S_{i+1}}$ of the next shot S_{i+1}, small amount of processing time is incurred.

3.1 Exploiting MPEG Motion

The types of motion vectors in MPEG provide hints for similarity measure among three temporally adjacent frames. Denote $|f^k_{mv}|$ and $|bi^k_{mv}|$ as the numbers of forward and bidirectional motion vectors of kth frame respectively, kth frame is regarded as the last frame of a shot if.

$$\frac{|f^k_{mv}|}{Z} > \delta_k \qquad (15)$$

$$\frac{|f^{k+1}_{mv}| + |bi^{k+1}_{mv}|}{Z} < \delta_{k+1} \qquad (16)$$

where Z is the number of macroblocks in a frame, and δ_k, δ_{k+1} are threshold values. Notice that if kth frame is an I-frame, (15) is not applicable.

3.2 Feature Similarity Measure

We adopt methods in [6] to extract color and texture features directly from the DCT domain for similarity measures. The color similarity between two frames is based on Euclidean distance measure,

$$\begin{aligned}&Sim^c(L^{S_i}, F^{S_{i+1}}) \qquad (17) \\ &= -\sum_{x\in\{H,S,V\}} \{\sum_{i=O}^{N.-1} (h^L_x(i) - h^F_x(i))^2\}^{\frac{1}{2}}\end{aligned}$$

where h^L_x is the color histogram of the DC image of L^{S_i} in HSV space and N_x is the quantized levels. Texture features are computed based on the local variance of DCT coefficients. Since DCT compacts the image energy into the lower order coefficients, we only consider the first nine AC coefficients in zig-zag order, i.e., $DCT_{u,v}$ for $0 < u + v \leq 3$. The texture feature is

$$\sigma^2_{u,v} = \mathbf{E}[DCT^2_{u,v}] - \mathbf{E}^2[DCT_{u,v}] \qquad (18)$$

where $\sigma_{u,v}$ and $\mathbf{E}[DCT_{u,v}]$ are the standard deviation and expectation of $DCT_{u,v}$ over all DCT blocks respectively. Similarly, we employ Euclidean distance for texture similarity measure Sim^t.

We also consider the problems of illumination effect. We apply the histogram transformation technique to suppress the changes. Denote $\hat{F}^{S_{i+1}}$ as the histogram transformed image, the similarity measure is based on the frame-to-frame difference metric,

$$Sim^l(L^{S_i}, F^{S_{i+1}}) = |L^{S_i} - F^{S_{i+1}}| - |L^{S_i} - \hat{F}^{S_{i+1}}| \quad (19)$$

which is expected to yield large value if $L^{S_i}, F^{S_{i+1}}$ belong to the same shot but undergo global luminance transitions.

Subsequently, the probability of a camera break between two suspected shots S_i and S_{i+1} is given by

$$P(B|S_i, S_{i+1}) = \frac{1}{Z} \exp \{-Sim(L^{S_i}, F^{S_{i+1}})\} \quad (20)$$

where
$Sim(L^{S_i}, F^{S_{i+1}}) = \sum_{x \in \{c,t,l\}} Sim^x(L^{S_i}, F^{S_{i+1}})$ and Z is a normalizing constant which can be ignored during the actual implementation.

4 Experimental Results

To evaluate the performance of the proposed approach, we conduct experiments on news sequences, documentary films, movies and TV streams. For simplicity, we define V_{cut} as the number of correctly detected camera cuts; V_{wipe} as the number of correctly detected wipe sequences; V_f as false alarms; V_m as missed detection; and V_n as the number of frames in a video.

4.1 Camera Cut Detection

Table 1 shows the experimental results of the proposed camera cut detection. On average, the spatio-temporal image model retains 5% of total frames for further shot pruning. The tested MPEG sequences: *cnn.mpg* involves camera flushing; *pearl.mpg* has fast camera and object motions; *sacrifice.mpg* involves fast object motion; *TungNien.mpg* and *SanGuo.mpg* consist of both indoor and outdoor long-take-shots with moderate object motions; *Shuahan.mpg* is an action sequence with special cinematographical effects; *ToWest.mpg* has rich of fighting and magical scenes.

Video	Type	V_n	V_{cut}	V_f	V_m
cnn.mpg	news	1980	10	2	0
pearl.mpg	news	4300	30	1	0
sacrifice.mpg	movie	738	5	0	0
TungNien.mpg	movie	11247	17	1	0
SanGuo.mpg	TV	5252	37	0	0
ShuShan.mpg	movie	9150	107	41	11
ToWest.mpg	TV	18954	96	12	5

Table 1: Camera cut detection results.

The experimental results show that the proposed approach performs stably for long-take shots, and the only false alarm in *TungNien.mpg* is due to noise generated when encoding the raw video data to MPEG file. Also, the approach works satisfactorily for fast action movies in the sense that it is able to parse some busy shots without causing false alarms. Figure 7 and 8 show some successful and failure examples. False alarms are mainly due to the sudden appearance of large moving objects across screen, and local illumination effect. Missed detections are because of low color-texture contrast between two adjacent shots.

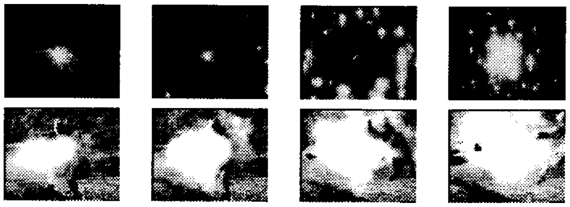


Figure 7: Success examples: no false alarms.

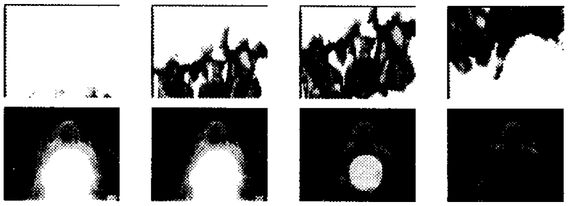


Figure 8: Fail examples: false alarms.

4.2 Camera Wipe Detection

In this experiment, we randomly select the shots detected in section 4.1 to synthesize new MPEG video sequences. These shots are connected by camera wipes. The wipe range is about 40 frames depending on the size of a video frame. Table 2 shows the experimental results of our proposed wipe detection approach on the sequences. The missed detections are due to the low image contrast between two connected shots. Figure 9 shows the spatio-temporal images of the detected and missed camera wipes. For all detected wipes, the number of undetected wipe frames are at most two frames.

4.3 Camera Break Detection

In this section, we activate both cut and wipe detectors to demonstrate the tolerance of the proposed approach to false and missed detections. Table 3 shows

Video	Type		V_n V_{wipe}	V_f	V_m
pearl.mpg	news	2000	9	0	3
ShuShan.mpg	movies	2280	16	0	2

Table 2: Experimental results of camera wipe detection.

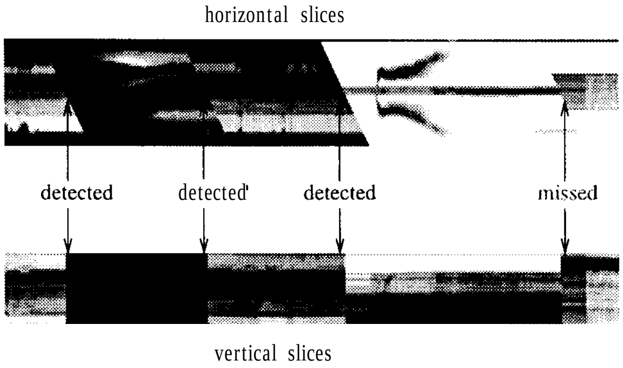


Figure 9: Camera wipe detection.

the experimental results. Our approach achieves accuracy close to 100% for cut detections; causes false alarms if a particular region in the spatio-temporal image resembles a wipe pattern.

	bahamas.mpg	malaysia.mpg
V_n	10177	12445
V_{cut}	43	62
V_f	1	1
V_m	1	0
V_{wipe}	2	0
V_f	1	0
V_m	0	0

Table 3: Camera break detections on documentary films.

5 Conclusion

We have proposed methods on detecting camera cuts and *wipes* in MPEG domain. Our methods reduce video segmentation to image segmentation problems and thus detect camera breaks with significant speed up. Shot pruning can be easily implemented by investigating motion vectors and DCT coefficients. In future, we will also develop dissolve detector based on the proposed spatio-temporal image model.

Acknowledgment

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

References

[1] A. M. Alattar, "Wipe Scene Change Detector for Use with Video Compression Algorithms and MPEG-7," *IEEE* Transactions on Consumer Electronics, vol. 44, no. 1, pp. 43-51, Feb 1998.

[2] D. Bordwell & K. Thompson, Film *Art: an Introduction,* Random House, 1986.

[3] A. Jain & F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters," Pattern Recognition, Vol 24, no. 12, 1991.

[4] W. Kong, Y. Ren & H. Lu, "A New Scene Breakpoint Detection Algorithm Using Slice of Video Stream," *IAPR* International Workshop on Multimedia *Media* Information Analysis and Retrieval, pp. 174-180, 1998.

[5] H. -C. H Liu, G. L. Zick, "Scene Decomposition of MPEG Compressed Video, " Digital Video Compression: Algorithms and Technologies, vol. 2419, pp. 26-37, 1995.

[6] C. W. Ngo, T. C. Pong & R. T. Chin, "Exploiting Image Indexing Techniques in DCT domain," IAPR International Workshop on Multimedia Media Information Analysis and *Retrieval,* pp. 195-206, 1998.

[7] N. V. Patel, I. K. Sethi, "Compressed Video Processing for Cut Detection," *IEE Proc.* Visual Image Sig*nal* Process, vol. 143, no. 5, pp. 315-23, Oct 1996.

[8] Bo Shen & Donger Li, Ishwar K. Sethi, "HDH Based Compressed Video Cut Detection," *Second Intl. Conf. on* Visual *Information Systems,* pp. 149-156, Dec 1997.

[9] B. L. Yeo and B. Liu, "On the Extraction of DC Sequence from MPEG Compressed Video," *IEEE Int. Conf. on Image* Processing, vol. 2, pp. 260-30, Oct 1995.

[10] W. Xiong & C. M. Lee, "Efficient Scene Change Detection and Camera Motion Annotation for Video Classification, " Journal *of* Computer Vision and Image Understanding, vol. 17, no. 2, pp. 161-81, 1998.

[11] H. J. Zhang, A. Kankanhalli, & S. W. Smoliar, "Automatic Partitioning of full-motion video," *ACM Multimedia* System, Vol. 1, No. 1, pp. 10-28, 1993.