Detection of Gradual Transitions through Temporal Slice Analysis

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 present approaches for detecting camera cuts, wipes and dissolves based on the analysis of spatio-temporal slices obtained from videos. These slices are composed of spatially and temporally coherent regions which can be perceived as shots. In the proposed methods, camera breaks are located by performing color-texture segmentation and statistical analysis on these video slices. In addition to detecting camera breaks, our methods can classify the detected breaks as camera cuts, wipes and dissolves in an efficient manner.

1 Introduction

A video is physically formed by shots; a shot is an uninterrupted segment of screen time, space and graphical configurations. The boundary between two shots is called camera break. 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 boundary line crossing the screen such that one shot gradually replaces another; a dissolve superimposes two shots where one shot gradually lighten while the other fade out slowly. Wipe and dissolve are normally referred to as *gradual transitions*.

In the current literature, there are various algorithms for detecting camera breaks [7, 10], in general, we can categorize them as statistic-based, histogram-based, feature-based, transformed-based, and motion-based. Most existing algorithms can segment a video sequence into shots correctly if the sequence has smooth frame transitions within a shot and abrupt spatial changes between shots. These algorithms are insensitive to gradual transitions since the change between two consecutive frames is small. Furthermore, methods based on the frame-to-frame difference metrics and feature analysis are computationally expensive. Although spatial and temporal sub-sampling of video frames are suggested to improve processing efficiency [8], 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.

Recently, we have presented works on detecting camera cut and wipe based on the analysis of *two* orthogonal slices [6]. The proposed video slicing is equivalent to the conventional spatial sampling of video frames, however, with the capabilities of revealing interesting visual cues when these slices are cascaded over times to form spatio-temporal images. These cues include the different patterns of region boundaries created by cuts and gradual transitions. By exploiting the patterns, we can bridge the dichotomy between detection and classification of camera breaks. In this paper, we improve the previous results by using *three* slices to correct the deficiency of missed detection. In addition, we reformulate the existing dissolve detector [1] and propose a slice voting scheme for dissolve detection.

2 The Concept of Video Slices

A slice is a $1D$ image taken from a frame; a spatio-temporal image is a collection of slices in the sequence at the same positions. Figure 1 shows three spatio-temporal images that are taken for analysis. To save computations, the images are extracted from DC images of I-frames, and the estimated DC images of P-frames and B-frames [9].

Denote f_{dc} as a $M \times N$ DC image, mathematically, our approach projects f_{dc} vertically, horizontally and diagonally to three $1D$ slices v, h and d,

$$v(i) = \sum_{p=k_1-j}^{k_1+j} \alpha_p f_{dc}(p,i), \text{ where } k_1 = \frac{M}{2} \quad (1)$$

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

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

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 the slices. To ensure the smoothness of slices within a shot, we set $j = 1$ and perform Gaussian smoothing on the slices, where $\alpha = [0.2236, 0.5477, 0.2336]$. By cascading these slices over time, we acquire a 2D image **V** formed by vertical slices, a 2D image **H** formed by horizontal slices, and a 2D image **D** formed by diagonal slices. Denote t as the time coordinate and (x, y) as the image coordinate, then **H**, **V** and **D** are in $x - t$, $y - t$ and $z - t$ space respectively.

Figure 2 shows the projected DC spatio-temporal images from MPEG videos. As seen in the figure, each image contains several spatially uniform texture regions, where each region is formed by the slices taken from frames that belong to a same shot. The type of camera breaks will affect the boundary shape of two connected regions. Figure 3 illustrates various patterns of spatio-temporal images. In general, a camera cut results in vertical boundary lines; a wipe results in slanted boundary lines; while a dissolve connects two regions slowly and does not have a clear boundary.

Based on this observation, we claim that the task of detecting camera breaks is equivalent to the task of segmenting image into regions; and therefore, we can reduce video segmentation problems to image segmentation problems. Furthermore, by investigating the orientation of boundary lines, we can classify cuts and wipes. Although dissolves do not create clear boundary lines, we can still apply statistical analysis to detect the breaks.

Previously proposed wipe detectors [2, 11] and dissolve detectors [1, 4] are based on statistical analysis, these approaches normally fail in detecting the exact begin and end frames of gradual transitions, in addition, suffers from difficulties in distinguishing wipes, dissolves and motions. In contrast, our wipe detector can identify the exact durations of wipe sequences as long as the two end points of slanted boundary lines are detected. A robust region segmentation algorithm can release the incapabilities in distinguishing wipes, dissolves and motions. Nevertheless, our dissolve detector also encounters problem on detecting the exact dissolve boundaries.

3 Camera Cut and Wipe Detection

In this section, we propose a Markov energy model to locate cuts and wipes based on the color and texture discontinuities happen at the boundaries of regions.

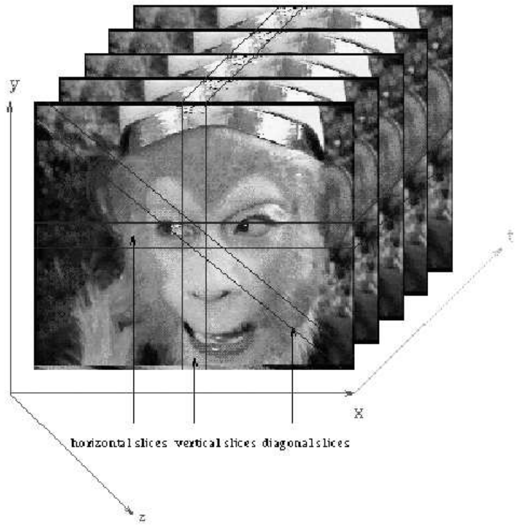


Figure 1: Three video slices taken from an image volume along the temporal dimension.

(a) Three shots connected by two cuts

(b) Two shots connected by a wipe

(c) Two shots connected by a dissolve

Figure 2: Samples of horizontal spatio-temporal images.

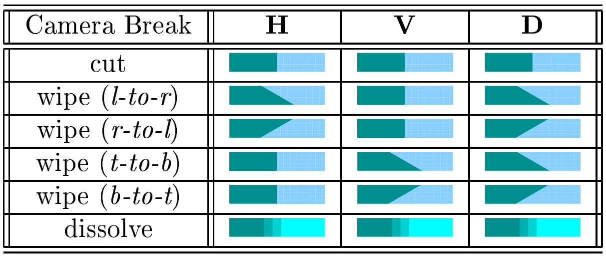


Figure 3: The image patterns created by camera breaks. Various wiping directions: *l-to-r* (left-to-right); *r-to-l* (right-to-left); *t-to-b* (top to bottom); *b-to-top* (bottom-to-top).

3.1 Feature Computing

Denote $\mathbf{H} = [H_r, H_g, H_b, H_y]$, $\mathbf{V} = [V_r, V_g, V_b, V_y]$ and $\mathbf{D} = [D_r, D_g, D_b, D_y]$ as the spatio-temporal images in (r, g, b) color space[1] to and y luminance space. The approach computes edge information by

$$E^{H_i}_{\sigma,\theta} = \mathbf{G\bar{D}}_{\sigma,\theta} * H_i \tag{4}$$

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

$$\begin{aligned} \mathbf{G\bar{D}}_{\sigma,\theta}(x,y) &= -\frac{x}{\sigma^2}\bar{\mathbf{G}}_{\sigma,\theta}(x,y) \\ \bar{\mathbf{G}}_{\sigma,\theta}(x,y) &= \bar{\mathbf{G}}_{\sigma}(x^{'}, y^{'}) \end{aligned} \tag{5}$$

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

The texture feature is computed based on Gabor decomposition [5]. The idea is to decompose images into multiple spatial-frequency channels, and use the real components of channel envelopes to form a feature vector. The complex Gabor images are,

$$T_{\sigma_x,\sigma_y,\theta} = \hat{\mathbf{G}}_{\sigma_\mathbf{x},\sigma_\mathbf{y},\theta} * H_y \tag{6}$$

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,

$$\hat{\mathbf{G}}_{\sigma_\mathbf{x},\sigma_\mathbf{y}}(x,y) = (\frac{1}{2\pi\sigma_x\sigma_y})\exp\{-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})\}\exp\{2\pi jWx\} \tag{7}$$

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

[1]Note that MPEG uses YCrCb color space; our method converts the YCrCb to RGB components

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.

3.2 Image Segmentation

We employ Markov energy model to describe the contextual dependency of spatio-temporal images for segmentation purpose. The probability that a pixel triple $\eta = (\eta_h, \eta_v, \eta_d)$ at $\mathbf{H}(k,t)$, $\mathbf{V}(k,t)$ and $\mathbf{D}(k,t)$ is on the region boundary ξ of two connected regions is

$$p(\eta \in \xi | \mathbf{H}, \mathbf{V}, \mathbf{D}) = p(\eta \in \xi | \mathbf{H}_N, \mathbf{V}_N, \mathbf{D}_N) \tag{8}$$

where $\mathbf{H}_N$, $\mathbf{V}_N$ and $\mathbf{D}_N$ are a 3×3 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.

V_1	V_2	V_3
V_4	x	V_6
V_7	V_8	V_9

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

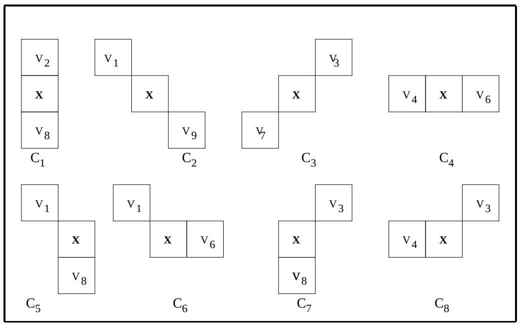


Figure 5: The connected components defined on the neighborhood system.

Assuming $\mathbf{H}_N$, $\mathbf{V}_N$ and $\mathbf{D}_N$ are independent, we rewrite (8) as,

$$p(\eta | \mathbf{H}, \mathbf{V}, \mathbf{D}) = p(\eta_h | \mathbf{H}_N) p(\eta_v | \mathbf{V}_N) p(\eta_d | \mathbf{D}_N) \tag{9}$$

and $p(\eta_h)$, $p(\eta_v)$, $p(\eta_d)$ are the probability of η_h, η_v η_d on a region boundary of $\mathbf{H}$, $\mathbf{V}$ and $\mathbf{D}$ respectively. By

Markov-Gibbs equivalence, we have

$$p(\eta_i) = \frac{1}{Z} \exp\{-U(\eta_i)\} \qquad (10)$$

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

$$U(\eta_i) = \sum_{c \in C} \beta_c \Gamma_c(\eta_i) \qquad (11)$$

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

$$\begin{bmatrix} U^r_{cut}(\eta^r_h) \\ U^r_{wipe^-}(\eta^r_h) \\ U^r_{wipe^+}(\eta^r_h) \end{bmatrix} = 3 \begin{bmatrix} \Gamma^r_{C_1}(\eta^r_h) \\ \Gamma^r_{c'}(\eta^r_h) \\ \Gamma^r_{c''}(\eta^r_h) \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Gamma^r_{C_1}(\eta^r_h) \\ \Gamma^r_{C_2}(\eta^r_h) \\ \Gamma^r_{C_3}(\eta^r_h) \end{bmatrix} - \begin{bmatrix} \Gamma^r_{C_4}(\eta^r_h) \\ \Gamma^r_{C_4}(\eta^r_h) \\ \Gamma^r_{C_4}(\eta^r_h) \end{bmatrix} \qquad (12)$$

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

$$\begin{aligned} \Gamma^r_{C_i}(\eta^r_h) &= \sum_\theta \{|E^{H_r}_{\sigma,\theta}(k,t) - E^{H_r}_{\sigma,\theta}(k_{\eta_1}, t_{\eta_1})| + \\ & \qquad |E^{H_r}_{\sigma,\theta}(k,t) - E^{H_r}_{\sigma,\theta}(k_{\eta_2}, t_{\eta_2})|\} \end{aligned} \qquad (13)$$

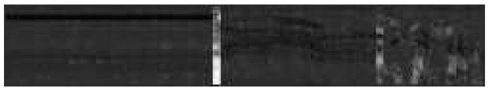
$\Gamma^g_{C_i}$ and $\Gamma^b_{C_i}$ are computed in a similar way; and $\Gamma^y_{C_i}$ is computed by

$$\begin{aligned} \Gamma^y_{C_i}(\eta^y_h) &= \sum_\theta \{|T_{\sigma_x,\sigma_y,\theta}(k,t) - T_{\sigma_x,\sigma_y,\theta}(k_{\eta_2}, t_{\eta_2})| + \\ & \qquad |T_{\sigma_x,\sigma_y,\theta}(k,t) - T_{\sigma_x,\sigma_y,\theta}(k_{\eta_1}, t_{\eta_1})|\} \end{aligned} \qquad (14)$$

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

$$U_{cut}(\eta_h) = \alpha_c \max_{j \in \{r,g,b\}} U^j_{cut}(\eta^j_h) + \alpha_t U^y_{cut}(\eta^y_h) \qquad (15)$$

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



(a) U_{cut} of horizontal slices

(b) U_{wipe^-} of horizontal slices

Figure 6: Segmentation results for images shown in Figure 2.

4 Dissolve Detection

A dissolve connects the boundaries of two shots smoothly; as a result, the connected shots share a smooth boundary region in the spatio-temporal image. Globally the image is composed of two regions with different visual surface; locally they exhibit a smooth transition from one region to another. Our goal is to segment the image into three portions: two regions representing successive shots and one narrow region representing the dissolve duration.

Denote $D(x, y, t)$ as the intensity function of frames superimposed by two shots having intensity functions $S_1(x, y, t)$ and $S_2(x, y, t)$ respectively. Suppose $D(x, y, t)$ starts at t_1 and ends at t_2, then

$$\begin{cases} S_1(x,y,t) & t < t_1 \\ (1-\alpha(t))S_1(x,y,t) + \alpha(t)S_2(x,y,t) & t_1 \le t \le t_2 \\ S_2(x,y,t) & t > t_2 \end{cases} \qquad (16)$$

where $\alpha(t) = \frac{t - t_1}{t_2 - t1}$ varys linearly with t in the range $[0, 1]$. Denote $\mu_i(t)$ be the mean intensity of a slice during the interval $t_1 < t < t_2$ in a spatio-temporal image i, then

$$\mu_i(t) = \mu_i^{S_1} + (\mu_i^{S_2}(t) - \mu_i^{S_1}(t))\alpha(t) \qquad (17)$$

where $\mu_i^{S_j}$ is the mean intensity of a slice that belongs to shot j and $i \in \{\mathbf{H}, \mathbf{V}, \mathbf{D}\}$. Taking the first derivative $\mu_i^{'}(t) = \frac{d\mu_i(t)}{dt}$, we have

$$\mu_i^{'}(t) = \frac{\mu_i^{S_2}(t) - \mu_i^{S_1}(t)}{t_2 - t_1} \tag{18}$$

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

Similarly, let $\sigma_i(t)$ be the variance of a slice during a dissolve in a spatial temporal image i, then

$$\begin{aligned} &\sigma_i(t) \\ &= \quad (\sigma_i^{S_1}(t) + \sigma_i^{S_2}(t))\alpha^2(t) - 2\sigma_i^{S_1}(t)\alpha(t) + \sigma_i^{S_2}(t) \end{aligned} \tag{19}$$

where $\sigma_i^{S_j}(t)$ is the variance of a slice that belongs to shot j and $i \in \{\mathbf{H}, \mathbf{V}, \mathbf{D}\}$. If $\sigma_i^{S_1}(t)$ and $\sigma_i^{S_2}(t)$ remain constant, $\sigma_i(t)$ is a concave upward parabola during $t_1 \leq t \leq t_2$.

The proposed dissolve detection algorithm computes (18) and (19) of a spatio-temporal image, and then records the periods that have approximately constant mean values and concave upward parabola curves for $15 \leq t_2 - t_1 \leq 45$. The assumptions in (18) and (19) will be seriously violated if there are vigorous motions during dissolves; however, in most cases dissolves involve only still to moderate motions. Under this scenario, we employ a voting scheme where a frame $f(t) \in D(x, y, t)$ if

$$\sum_{i \in \{\mathbf{H}, \mathbf{V}, \mathbf{D}\}} G(\sigma_i(t), \mu_i^{'}(t)) \geq 2 \tag{20}$$

where $G : \mathbb{R} \times \mathbb{R} \rightarrow \{0, 1\}$ is a logical operator.

5 Experimental Results

We conduct experiments to evaluate the performance of the proposed methods. The tested videos consist of slow to fast camera motions, fast and large moving objects. A wipe spans about 40 frames; while a dissolve crosses about 30 frames.

Tested Video	Cut			Wipe			Dissolve		
	D	M	F	D	M	F	D	M	F
syn.mpg	5	0	0	7	1	0	5	0	0
ba.mpg	45	1	0	2	0	1	4	2	2
gf2.mpg	44	3	0	0	0	4	8	9	3
gf3.mpg	51	3	0	0	0	0	4	0	2
recall	0.95			0.90			0.54		
precision	1.00			0.62			0.71		

Table 1: Camera break detection results. D: correct detections; M missed detections; F: false alarms.

Table 1 shows the experimental results of the proposed detection methods. For all correctly detected wipes and dissolves, at least 10 frames of the actual sequences are covered. To investigate the tolerance and accuracy of gradual transitions detection, we further perform recall-precision to evaluate the results. Denote A_i as the number of frames due to action i; B_i as the number of detected frames in class i; C_i as the number of correctly detected frames in class i, then

$$recall_i = \frac{C_i}{A_i} \tag{21}$$

$$precision_i = \frac{C_i}{B_i} \tag{22}$$

where $i \in \{cut, wipe, dissolve\}$, $recall_i$ and $precision_i$ are in the interval of $[0, 1]$. Low recall values indicate the frequent occurrence of missed detections, while low precision show the frequent occurrence of false alarms.

Through the experiments, our cut detector[2]. achieves approximately 0.95 and 1.00 for recall and precision measures respectively. The missed cuts are due to low texture contrast between two adjacent shots. The wipe detector can locate most of the wipe sequences, however, suffers from false detection if a particular region in the spatio-temporal image resembles a wipe pattern. The false alarms may be further pruned by examining the existence of a moving line through the first and second AC coefficients of suspected wipe frames. Figure 7 further shows the spatio-temporal images of the detected and missed camera wipes. The miss detection is due to the unclear vertical region boundary in the vertical slices. The dissolve detector is sensitive to the underlying motions of a sequence; the detector can robustly detect dissolve sequences of slight or no motions (see Figure 8 for an illustration), however, will result in missed detections if the motions violate the distribution shapes of equations (18) and (19).

6 Conclusion

We have proposed methods on detecting and classifying camera *cuts*, *wipes* and *dissolves* based on the analysis of video slices. Our methods reduce the video segmentation problems to image segmentation problems, in additions, process frames directly in MPEG domain, resulting in significant speed up. In contrast to most of the previous empirical studies which detect *any arbitrary frame* in a wipe or dissolve as a break point, we detect *a shot sequence* as a break point; and

[2]The cut detector will also perform pruning by investigating the DCT coefficients and motion vectors of suspected cut frames (about 5% of total frames). The detail is presented in [6].

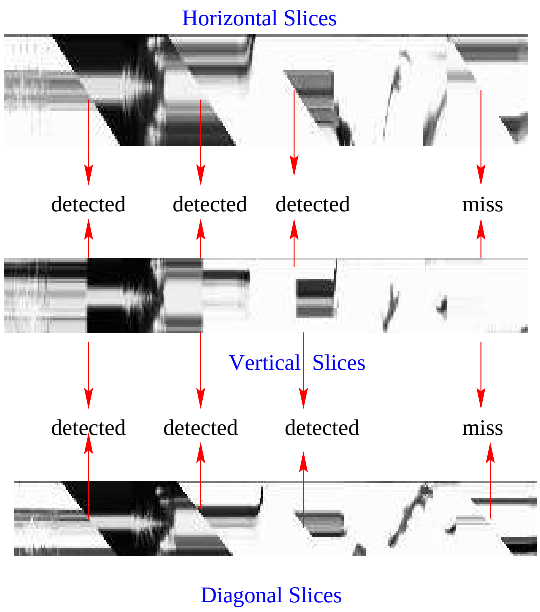


Figure 7: Camera wipe detection.

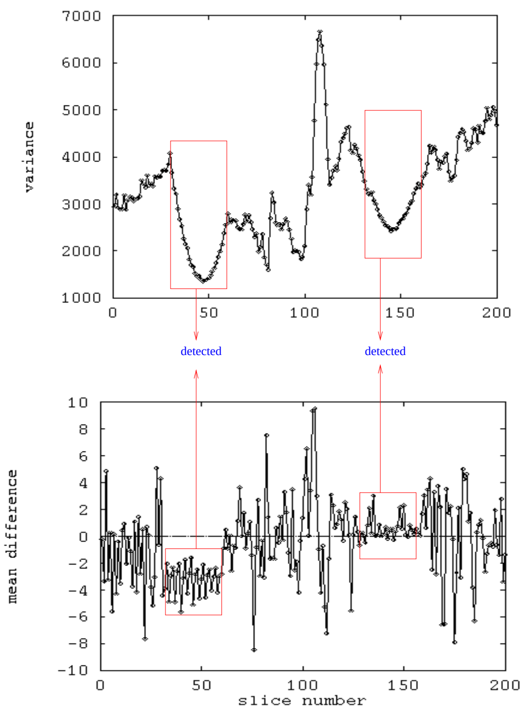


Figure 8: Analysis of dissolve sequences of slight motion on a horizontal spatio-temporal image.

hence, further perform recall-precision analysis to ratify the accuracy. In future, we will study the possibility of estimating image and motion features directly from slices for video database indexing and retrieval.

Acknowledgements

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

References

[1] A. M. Alattar, "Detecting and Compressing Dissolve Regions in Video Sequences with a DVI Multimedia Image Compression Algorithm," *IEEE Intl. Symposium on Circuits and Systems*, vol. 1, pp.13-16, 1993.

[2] 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.

[3] D. Bordwell & K. Thompson, *Film Art: an Introduction*, University of Wisconsin, 2nd edition, Random House, 1986.

[4] L. Gu, K. Tsui, & D. Keightley, "Dissolve Detection in MPEG Compressed Video," *IEEE Int. Conf. on Intelligent Processing Systems*, pp. 1692-1696, 1997.

B. Jähne, *Spatio-Temporal Image Processing*, Springer-Verlag Berlin Heidelberg 1993.

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

[6] C. W. Ngo, T. C. Pong & R. T. Chin, "Camera Breaks Detection by Partitioning of 2D Spatio-temporal Images in MPEG Domain," *Int. Conf. on Multimedia Computing and Systems*, 1999, to appear.

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

S. L. Peng, "Temporal Slice Analysis of Image Sequences," *Proc. on Computer Vision and Pattern Recognition*, pp. 283-8, 1991.

[8] W. Xiong & C. M. Lee, "Efficient Scene Change Detection and Camera Motion Annotation for Video Classification, " *Journal of Computer Vision and Image Understanding*, vol. 71, no. 2, pp. 166-81, 1998.

[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. 260-3O, Oct 1995.

[10] 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.

[11] M. Wu, W. Wolf & B. Liu, "An Algorithm for Wipe Detection," *IEEE Intl. Conf. on Image Processing*, 1998.