# DETECTION OF VARIANT WIPE EFFECTS

*Shan Li, M.C. Lee*

Department of Computer Science & Engineering, the Chinese University of HK, Hong Kong
{sli, mclee}@cse.cuhk.edu.hk

## ABSTRACT

Due to the diversity of different wipe effects, wipe transition is considered complex and difficult to detect. This paper identifies two common characteristics of different wipes, which can be described by two principles – independence and completeness. By exploiting the two principles, we developed an effective wipe detector. Since no pre-determined values would be appropriate for different videos, we propose using a dynamic method to generate adaptive thresholds for wipe detection. The experimental results show that the proposed detection method can identify various wipe effects with good accuracy; overall, it outperformed other published methods.

## 1. INTRODUCTION

High-level video processing techniques require accurate video segmentation, which temporally segments a video sequence into adjacent shots. Meanwhile, transition types between the two consecutive shots need to be detected. The transitions between any two consecutive shots can be grouped into two types, namely, abrupt change, and gradual transitions (dissolve, wipe, fade, etc.). Wipe transition is an important type of gradual transition, which is extensively used in video programs. About 24 different wipe effects are commonly used in video editing. Compared to dissolve transitions, wipes are more difficult to detect and less discussed in previous work, possibly due to the wide variety of wipe effects. Fig.1 illustrates a wipe effect example.

Fig.1. A band wipe effect example

A number of researches have been done on wipe detection. Some work [1][5] developed wipe detectors based on the statistical model of the wipe frames. Yet their method has a high false alarm rate. Wu et al. [10] proposed a wipe detection method based on the standard deviation of projected pixel-wise difference. It can only detect a limited number of wipe effects. More recent approaches [3][8] detect wipe effects by investigating the orientation of boundary lines in the spatial temporal image. They can only detect the wipe effects that produce slanted line in spatial temporal images. Pei and Chou [9] proposed to detect wipes on MPEG stream by accumulating macro blocks with content changes. Yet, the motion vectors produced by MPEG encoders may not reflect the true mapping relation between two blocks, not to mention that some MPEG streams do not contain motion vectors at all. Other approaches [2][4] use the center and variance of the change regions to characterize the pattern of wipe effect. Obviously, this method can only characterize a few types of simple wipe effects.

In this paper, we propose to develop a general detector that can detect various wipe effects. We first identify the common characteristics shared by different types of wipes. A wipe detector is then developed to exploit the satisfaction of the two principles in a candidate wipe sequence.

The rest of the paper is organized as follows. Section 2 introduces the common characteristics of different types of wipes. Section 3 proposes a new wipe detection method. Experiment results and analysis are presented in section 4. Section 5 concludes the paper.

## 2. WIPE CHARACTERIZATION

During a wipe transition, the end frame gradually shows up and wipes out the start frame. A part of the start frame will be replaced by the corresponding part of the end frame when the current frame proceeds to the next one along the wipe sequence. The changing part between two consecutive frames refers to the part being replaced. It can be obtained by comparing the two consecutive frames in the transition sequence. Suppose the wipe transition spread between frame 1 and frame N. Let the changing part (i.e., the collection of pixels whose color values are changed) between frame t and frame t+1 be denoted by $\xi_t$ ($1 \le t \le N-1$). The start frame belongs to the source shot ($S_1$) and the end frame belongs to the destination shot ($S_2$). Then we can model an ideal wipe sequence (i.e., during an ideal wipe transition, the two shots are assumed to be motionless) as:

$$S(x,y,t) = \begin{cases} S_2(x,y) & \forall (x,y) \in \xi_1 \cup \xi_2 \cup \ldots \cup \xi_{t-1} \quad 1 \le t \le N-1 \\ S_1(x,y) & \text{otherwise} \end{cases} \qquad (1)$$

$S(x,y,t)$ is the value of pixel $(x, y)$ at frame $t$. $N$ is the total number of frames in the wipe transition. $\xi_t$ is defined as:

$$\xi_t = \{(x,y) \mid S(x,y,t) \ne S(x,y,t+1)\} \qquad (2)$$

In an ideal wipe sequence, two important observations can be made:

1) Any two elements $\xi_i$ and $\xi_j$ from $\xi = \{\xi_t \mid t = 1,...N-1\}$ (see Eq. 2) should not overlap with each other (i.e., $\xi_i$ and $\xi_j$ are mutually exclusive). This characteristic is referred to as the independence of the elements in the set $\xi$.

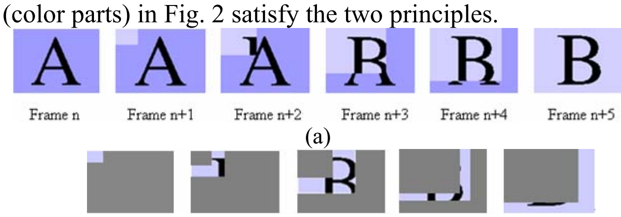
2) The union of all the elements in the set of $\xi$ should cover the entire frame. This characteristic is referred to as the completeness of the elements in the set of $\xi$.

Statement 1 indicates that in an ideal wipe transition, any pixel will only change once its color value during the wipe transition. Statement 2 means that all the pixels will have their color values changed after the completion of the wipe transition. When both the completeness and independence property of the set $\xi$ are met for a video sequence, a wipe transition can be declared.

Note that the principle of completeness is valid based on the assumption that $S_1(x,y)$ and $S_2(x,y)$ are uncorrelated, based on the observation that in a practical video application, the two involved shots usually display distinct color layouts.

The principle of independence is valid based on the assumption that both $S_1(x,y)$ and $S_2(x,y)$ have no motions. We noticed that during a wipe transition, the two involved shots are seldom associated with fast motions, since otherwise it will be quite difficult for the audience to tell the wipe effect clearly. Therefore we could apply block-based techniques to calculate $\xi$ to partially overcome the noise of minor motions. Adaptive thresholds can further be applied to improve $\xi$'s robustness. Details will be shown in section 3.

Fig. 2 gives an example of the changing parts during a wipe. The color/gray parts correspond to the collections of pixels that have/have no value change between any two consecutive frames respectively. Clearly, the changing parts (color parts) in Fig. 2 satisfy the two principles.




(a)

(b)

Fig.2. (a) A sample wipe sequence; (b) The changing parts between consecutive frames

## 3. WIPE DETECTION PROPOSAL

In this section, we propose a wipe detection method using the two principles we have found in the previous section.

Intuitively, the changing part $\xi$ can be calculated based on pixel-wise comparison of two consecutive frames (see Eq.2). Since minor motions are allowed in real video applications, we calculate $\xi$ based on block-wise comparison. DC coefficient of the block is used as the feature here since it can be readily obtained from an MPEG stream [6].

Each pixel in a DC frame is the DC coefficient of the 8×8 block in the original uncompressed image frame. The changing part between DC frames $t$ and $t$+1 is noted as $\xi_t^{'}$ ($1 \le t \le N-1$):

$$\xi_t^{'} = \{(x,y) \mid \|S_{dc}(x,y,t) - S_{dc}(x,y,t+1)\| \ge \delta_1 \mid\} \quad (3)$$

Where $\|\cdot\|$ indicates the absolute value; $S_{dc}(x,y,i)$ denotes the color value of pixel $(x,y)$ at the $t$-th DC frame. The threshold $\delta_1$ indicates the minimum change a pixel should have in order to be considered as a "changing pixel"; $\delta_1$ is determined adaptively, to improve $\xi_t^{'}$'s motions robustness:

$$\delta_1 = \sqrt{(\sum_x \sum_y [S_{dc}(x,y,t) - S_{dc}(x,y,t+1)]^2)/(N_x N_y)} \quad (4)$$

$N_x, N_y$ are the size of the DC frame.

A map $\Gamma$ is then built to check the independence and completeness of the set of $\xi^{'} = \{\xi_t^{'} \mid t = 1,...N-1\}$. The size of $\Gamma$ is the same as the size of the DC frame in the video sequence. $\Gamma$ will be updated from its initial value $\Gamma_0$ to $\Gamma_{N-1}$ using the elements from the set $\xi^{'}$. The value of each position $(x,y)$ in $\Gamma_0$ is zero; and $\Gamma_t$ can be updated from $\Gamma_{t-1}$ as:

$$\Gamma_t(x,y) = \begin{cases} \Gamma_{t-1}(x,y)+1 & (x,y) \in \xi_t^{'} \quad 1 \le t \le N-1 \\ \Gamma_{t-1}(x,y) & \text{otherwise} \end{cases} \quad (5)$$

Clearly, $\Gamma_{N-1}(x,y)$ indicates that the position $(x,y)$ have changed its color value $\Gamma_{N-1}(x,y)$ times during the wipe transition. It satisfies the relation: $0 \le \Gamma_{N-1}(x,y) \le N-1$.

For an ideal wipe effect, $\Gamma_{N-1}(x,y)$ should be equal to 1 for each position $(x,y)$. If $\Gamma_{N-1}(x,y) > 1$, the independence requirement would be violated; if $\Gamma_{N-1}(x,y) = 0$, the completeness requirement would be violated. We can then count the amount of violations in the map $\Gamma_{N-1}$. If the amount of violations is smaller than some threshold, a wipe effect can be declared.

To measure the amount of violations in $\Gamma_{N-1}$, we define respectively two cost functions $F_1$ and $F_2$ for violating the principles of independence and completeness as:

$$F_1(\Gamma_{N-1}) = \frac{1}{N_x N_y} \sum_{(x,y) \mid \Gamma_{N-1}(x,y) > 1} p(\Gamma_{N-1}(x,y)) \quad (6)$$

$$F_2(\Gamma_{N-1}) = \frac{1}{N_x N_y} \sum_{x,y \mid \Gamma_{N-1}(x,y)=0} p(0) \quad (7)$$

Eq. 6 and Eq.7 calculate penalty $p$ for each position and then use their mean value as the cost function value. The penalty function $p$ is defined as:

$$p(\Gamma_{N-1}(x,y)) = \begin{cases} 1, & \Gamma_{N-1}(x,y) = 0 \\ 0, & \Gamma_{N-1}(x,y) = 1 \\ \exp(\Gamma_{N-1}(x,y) - 1), & \Gamma_{N-1}(x,y) = 2,3,...,N-1 \end{cases} \quad (8)$$

In the case of independence violation, the penalty should be bigger when $\Gamma_{N-1}(x,y)$ is bigger, since there will be more violations of the independence principle. In our implementation, the penalty grows exponentially.

In order to detect various/generic types types of wipes, we propose a wipe detection algorithm that consists of three steps: 1) Extracting candidate wipe sequences; 2) Wipe detection within a given candidate video sequence; 3) Merging wipe sequences with small gaps. We describe the detail of these steps in the following.

A) *Extracting candidate wipe sequences*

Assuming that the cut boundaries are already found using the method reported in [7], we could use the histogram-based inter-frame differences, which are already known from the work [7], to prune the frames that are either still or have slow changes. The frames that have inter-frame differences bigger than $\delta_2$ for a sustained period (say, 10~24 frames) are regarded as candidate sequences. $\delta_2$ is set adaptively:

$$\delta_2 = \mu + \beta \cdot \sigma \,; \quad \delta_2 \ge 0 \tag{9}$$

$\mu$ and $\sigma$ are respectively the mean and standard deviation of the inter-frame differences between a given pair of cuts. $\beta$ controls the "tightness" of $\delta_2$.

B) *Wipe detection within a given candidate video sequence*

Within every candidate frame sequence, we detect wipes using the algorithm below. *L* is the total number of frames in the candidate sequence; *C_start* is the first frame number of the candidate sequence; *P_start* is the potential start frame number of a wipe; *C*(*j*-1,*j*) refers to the collection of changing blocks between frame *j*-1 and frame *j*; $\delta_3$ is the stop criteria. The detection algorithm is depicted as follows:

**Algorithm 1: Wipe detection within a candidate video sequence**

1 *P_start* ← *C_start*, *j* ← *C_start*, Initialize the map to $\Gamma_0$
2 **while** $j \le L + C\_start - 1$
3 *j* ← *j*+1; Compare frames *j* and *j*+1 to get *C*(*j*, *j*+1)
4 Update: $\Gamma_j \leftarrow \Gamma_{j-1} + C(j,j+1)$ //for every block in *C*(*j*,*j*+1), the value of its corresponding position in $\Gamma$ will increase by 1
5 **if** $F_1(\Gamma_j) > \delta_3$ //The updating process is stopped when the independence principle is violated at frame *j*
6 **if** $F_2(\Gamma_{j-1}) < \delta_4$ **and** $j - P\_start > 12$ //check if the completeness is met, and if the sequence is long enough
7 Declare a wipe, starting from *P_start*, ending at *j*-1
8 *P_start* ← *j*; $\Gamma_j \leftarrow \Gamma_0$
9 **else** Start wipe detection again from *P_start* +1
10 **end if; end if; end while**
11 **if** $F_1(\Gamma_{j-1}) > \delta_3$ **and** $F_2(\Gamma_{j-1}) < \delta_4$ **and** *j*-*P_start*>12 //in case that the principle of independence is never violated
12 Declare a wipe, starting from *P_start*, ending at *j*. **end if**

In the algorithm, $\Gamma$ is updated along the sequence until $F_1(\Gamma_j)$ exceeds the threshold $\delta_3$. If $F_2(\Gamma_{j-1})$ is smaller than a predefined threshold $\delta_4$, the frames between *P_start* and *j*-1 satisfy both the principles of independence and completeness, therefore forming a potential wipe sequence. If those frames are long enough (i.e., longer than 12 frames), it's then declared as a real wipe.

*C) Merging wipe sequences with small gaps*

After detecting wipes from all candidate sequences, we merge those wipe sequences with small gaps into a new one, since the small gaps(less than 3 frames) are more likely to be missed wipe frames.

## 4. EXPERIMENTS

We test the proposal on a number of video sequences. 24 edited MPEG-1 video sequences with 24 different wipe effects are included in our test data. 11 real MPEG-1 sequences, including 1 documentary, 4 movies, 6 ads, with overall 42 wipe effects inside, are also used as our test data.

Before applying the proposed wipe detection method to the video sequences, a set of thresholds/parameters ( $\delta_2$, $\beta$, $\delta_3$, $\delta_4$ ) should first be determined. They can be left to system user to adjust the trade-off between miss detection and false alarm rate and are experimented here.

We first adopt a threshold set { $\beta = 0$ , $\delta_3 = \exp(1)/2$ , $\delta_4 = 0.25$ } and the experiment results on the edited MPEG-1 video clips show that there are 0 miss and 2 false alarms. The 2 false alarms come from fast object motions within the frame and could be possibly removed by decreasing the threshold $\delta_3$ .Intuitively, tuning the thresholds could cause trade off between false alarm and miss detection. Experiments have been performed to investigate the impact of the thresholds/ parameters on detection performance. The results are shown in Table 1-3.

Table 1. Experiment results using different values of $\beta$ ( $\delta_3 = \exp(1)/2, \delta_4 = 0.25$ )

| $\beta$ | Detected | misses | false alarms |
|---|---|---|---|
| 0 | 24/24 | 0 | 2 |
| 0.2 | 24/24 | 0 | 0 |
| 0.5 | 20/24 | 4 | 0 |
| 1 | 19/24 | 5 | 0 |

Table 2. Experiment results for using different values of $\delta_3$ ( $\beta = 0.2, \delta_4 = 0.25$ )

| $\delta_3$ | Detected | misses | false alarms |
|---|---|---|---|
| exp(1)/4 | 16/24 | 8 | 0 |
| exp(1)/3 | 21/24 | 3 | 0 |
| exp(1)/2 | 24/24 | 0 | 0 |
| exp(1) | 24/24 | 0 | 4 |

As the experiment results in Table 1, 2 and 3 suggest, we can choose the values for the thresholds according to the system user's tolerance for false alarm and miss detection rate.

Using the set ( $\beta = 0.2$ , $\delta_3 = \exp(1)/2$, $\delta_4 = 0.25$ ), we compare the proposed method with other methods, including statistical method proposed in [1] (quoted as SM), the trajectory method proposed in [2] (quoted as TM), and the motion vector based method in [9] (quoted as MVM). We tuned the respective parameters of those methods so that the three methods achieve their optimal average performances in the 24 edited MPEG 1 sequences. The comparison results are listed in Tables 4 and 5 respectively.

Table 3. Experiment results for using different values of $\delta_4$ ( $\beta = 0.2, \delta_3 = \exp(1)/2$ )

| $\delta_4$ | Detected | misses | false alarms |
|---|---|---|---|
| 0.1 | 21/24 | 3 | 0 |
| 0.2 | 23/24 | 1 | 0 |
| 0.25 | 24/24 | 0 | 0 |
| 0.5 | 24/24 | 0 | 7 |

Table 4. Comparison on 24 edited wipe sequences

| Methods | Detected | Misses | False alarms |
|---|---|---|---|
| Our method | 24/24 | 0 | 0 |
| SM | 19/24 | 5 | 11 |
| TM | 4/24 | 20 | 4 |
| MVM | 22/24 | 2 | 3 |

Table 5. Comparison on 11 real video sequences

| Methods | detected | misses | false alarms |
|---|---|---|---|
| Our proposal | 39/42 | 3 | 5 |
| SM | 36/42 | 6 | 54 |
| TM | 14/42 | 28 | 3 |
| MVM | 35/42 | 7 | 19 |

For the edited sequences with little noise, Table 7 indicates that our proposed method can detect all wipe effects effectively. Note that the dissolves and camera/object motions present in the edited video sequences were not detected as wipes by our method. On the contrary, SM raised a large number of false alarms, which could have been caused by the intensive object/camera motions in the videos. MVM also produced a high rate of false alarms. TM produced the worst performance in terms of the number of misses, possibly because it could detect only several types of wipe effects. Note that MVM only checks for the principle of completeness, which accounts partially for its weak performance, since fast motions and other effects could also gradually change the contents of many blocks in the video frame, leading to false alarms. The non-precise motion vectors in the compressed domain could also cause weak performance.

The results for the real video sequences are as presented in Table 5. Among the 42 wipe transitions, our method detected 39 of them and had 5 false alarms. It also managed to reduce the false alarms generally caused by other types of gradual transitions and camera/object motions. Through a close examination of the 2 missed wipes, we note that there are very fast object/camera motions in the two involved shots. In the case of fast motions, our proposed method can detect a great number of changing regions between every two consecutive frames; so the principle of independence could be violated quickly during the detection process.

## 5. CONCLUSION

This paper proposes an effective method, which exploits two common properties of wipe effects, for detecting different types of wipe transitions – independence and completeness. Though the properties of wipes in a real video sequence may deviate from those of an ideal wipe, in practice, with the introduction of a penalty function, we can also apply the two principles to check satisfactorily for wipes in real videos. The experiment results show that the method can detect different wipe effects effectively.

The proposed method uses the block-based feature DC image to improve its robustness to small motions in the video sequence. Of course, besides DC image, other video frame features could also be used in our proposed wipe transition detection method. In our future work, we would search for more effective features which could be more robust to fast object motions.

**ACKNOWLEDGEMENT**

The work reported in this article has been supported in part by the Hong Kong Research Grants Council under CUHK4377/02E.

## 6. 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, Issue 1, pp.43 – 51,1998

[2] P. Campisi, A. Neri and, L. Sorgi, "Wipe effect detection for video sequences," *IEEE Workshop on Multimedia Signal Processing*, pp.161 – 164, Dec. 2002

[3] M.S. Drew, Ze-Nian Li, and Xiang Zhong, "Video dissolve and wipe detection via spatio-temporal images of chromatic histogram differences," *Proc. of IEEE International Conference on Image Processing*, vol.3, pp. 929 – 932, Sept. 2000

[4] H. Yu, G. Bozdagi and S. Harrington, "Feature-based Hierarchical Video Segmentation," *IEEE International Conference on Image Processing*, Vol. 2, pp. 498-501, 1997.

[5] W.A.C. Fernando, C.N. Canagarajah, and D.R. Bull, "Wipe Scene Change Detection in Video Sequences," *in Proc. ICASSP*, pp. 294-298, 1999.

[6] B.-L., Yeo and B. Liu, " Rapid Scene Analysis on Compressed video," *IEEE Trans. on Circuits and Systems for Video Technolo*gy, vol.5, No.6, pp. 533—544, 1995

[7] Shan Li and M.C.Lee, "An improved sliding window method for shot change detection," In Proceedings of the *7th IASTED International Conference on Signal and Image Processing*, Hawaii, Honolulu, pp. 246—254, Aug., 2005

[8] C.W. Ngo, T.C. Pong, and H.J. Zhang, "On Clustering and Retrieval of Video Shots Through Temporal Slices Analysis," *IEEE Transactions on Multimedia*, vol. 4, Issue 4, pp.446 – 458, Dec. 2002

[9] Soo-Chang Pei and Yu-Zuong Chou, "Effective wipe detection in MPEG compressed video using macro block type information," *IEEE Transactions on Multimedia*, vol.4, Issue 3, pp. 309 – 319, 2002

[10] M. Wu, W. Wolf, and B. Liu, "An Algorithm for Wipe Detection," *in Proc. ICIP*, pp. 893-897, 1998.