TRECVID2005 Experiments in The Hong Kong Polytechnic University: Shot Boundary Detection Based on a Multi-Step Comparison Scheme

Cheng Cai[a,b], Kin Man Lam*[a], Zheng Tan[b]

[a]Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong;

[b] School of Electronic and Information Engineering, Xi'an Jiaotong University, China.

enkmlam@polyu.edu.hk

Abstract

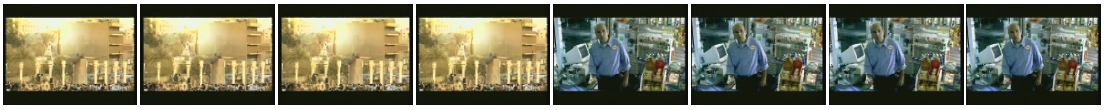
In this paper, we describe our experiments for TRECVID 2005 for the shot boundary detection task. Our approach is based on a multi-step comparison of the video frames. By measuring the difference between frames at varying distances apart, a distance map is generated and used to determine the existence of a transition and its type. The contents of the frames in a video shot are similar. If the difference between two consecutive frames is relatively large, a cut should happen. During a gradual transition, the difference between two consecutive frames is relatively small; therefore differences between the more distant frames are needed. While the comparative step size or the distance between two frames is equal to or larger than the length of transition, the difference between the frames during the transition will be much larger than that within the same video shot. In a distance map, a cut will appear as a triangle, a flash as two straight lines, and a gradual transition as a trapezoid. Based on the distance map, the different transitions can be detected and classified easily.

Keywords: Shot Boundary Detection, Cut Detection, Gradual Transitions, Video Partitioning, Video Analysis.

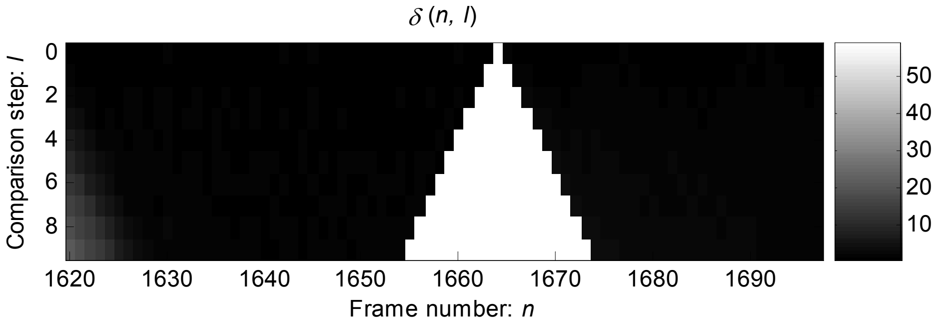
1 Introduction

A shot boundary can be categorized into cuts and gradual transitions. There are numerous algorithms for shot boundary detection, which are based on color (Yeo, and Liu 1995, Zhang, Kankanhalli, and Smoliar 1993), edge (Lienhart 1999, Zabih, Miller, and Mai 1999), motion (Bouthemy 1999), macroblock information (Pei, and Chou 1999, Jun, Yoon, and Lee 2000), variance curve (Alattar 1993, Ngo, Pong, and Chin 2001), linear regression (Han, and Kweon 2003), machine learning (Ngo 2003, Bescós 2004), statistical analysis (Hanjalic 2002, Vasconcelos, and Lippman 2000, Liu, Lo, Zhang, and Feng 2004), and linear prediction (Cai, Lam, and Tan 2004, 2005). The different approaches carry different assumptions regarding the detection of shot boundaries. For instance, the histogram-based method assumes that images from two different consecutive shots should have different histograms. The method based on the variance curve assumes that the variance between consecutive images in a video will not be zero. The assumption made by the linear prediction model is that the pixel values in a frame can be predicted from its previous frames, even if the frames are under some type of gradual transition such as dissolve. For cut detection, most of these methods have reasonable accuracy, because the difference between two consecutive frames is relatively large. However, gradual transition detection is a much more difficult problem, as the difference between successive frames under the transition is relatively small. Consequently, comparing the frames a certain step apart is necessary. With the development of software tools for video processing, gradual transition has become more complex than ever. The gradual transitions may have an arbitrary length of arbitrary type. Therefore, in our proposed method we compute the differences between frames with multi-steps to generate a distance map. Transitions are detected by analysing their patterns in the distance map.

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong SAR, China (Project No. PolyU 5220/03E).

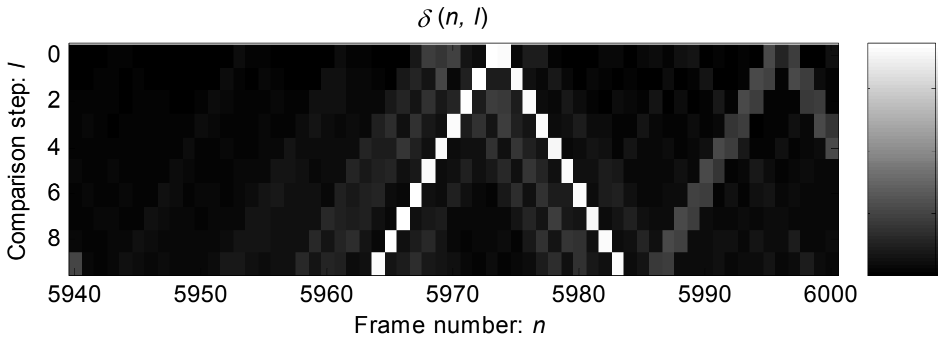


(a) Example of a cut.



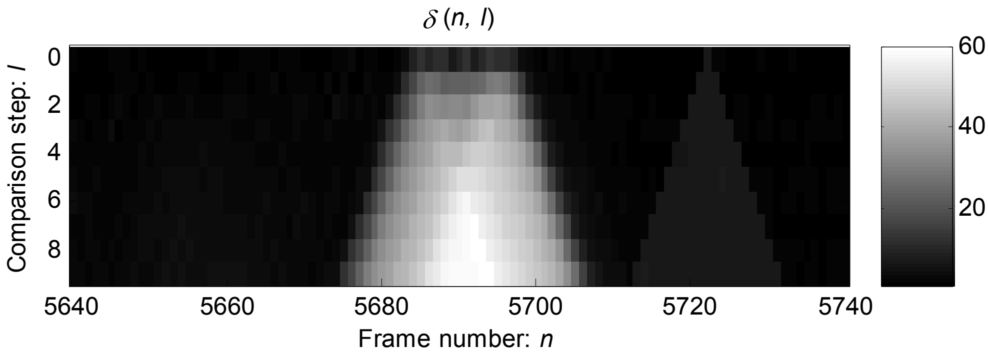
(b) $\delta(n, l)$ of a cut transition.

(c) Example of a flash.



(d) $\delta(n, l)$ of a flash.

(e) Example of a gradual transition.



(d) $\delta(n, l)$ of a gradual transition.

Figure 1. The Multi-step comparison $\delta(n, l)$ for a cut, a flash, and a gradual transition.

2 Comparing Frames at Multiple Steps

This section will describe our multi-step comparison methods for the detection of cuts and gradual transitions. We compare the frames based on histograms in the RGB colour space. To reduce the storage and computational requirements, each of the color components is quantized in 3 bits, so the length of a histogram is $2^3 \times 2^3 \times 2^3 = 512$.

In a video shot, histograms of the frames should be similar to each other. Suppose that $h(n, i)$, where $0 \le i \le M-1$, represents the colour histogram of the n^{th} frame, n and i denote the frame number in the video sequence and the bin number in the histogram, respectively. $M = 512$ because the color histograms used in our method has 512 bins. The distance map, which represents the differences between two frames at multi-steps apart, is defined as follows:

$$\delta(n,l) = \frac{100}{2 \cdot W \cdot H} \cdot \sum_{i=0}^{M-1} \left| h(n-l,i) - h(n+1+l,i) \right|, \quad (1)$$

where $h(n-l, i)$ and $h(n+1+l, i)$ denote the histograms of frames $n-l$ and $n+1+l$, respectively, $\delta(n, l)$ denotes the histogram difference between $h(n-l, i)$ and $h(n+1+l, i)$, and W and H denote the width and height of frame, respectively. The dynamic range of $\delta(n, l)$ is between 0 and 100. l is the step parameter which is the distance between two frames. If $l = 0$, $\delta(n, l)$ represents the difference between two consecutive frames. When l increases, long gradual transitions will more likely be detected. However, the possibility of a short video shot being misclassified will increase at the same time. In our system, we set $0 \le l \le 10$.

2.1 Pattern of a Cut

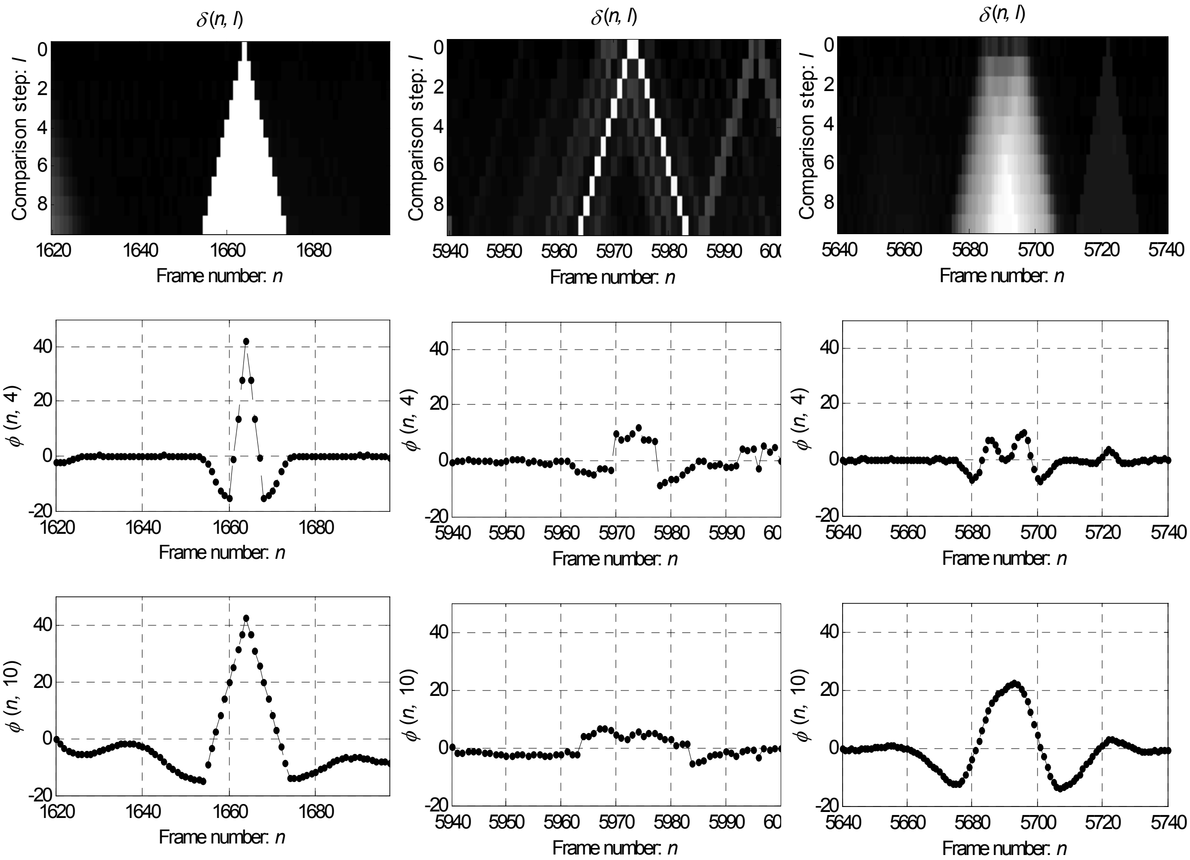
Suppose a cut occurs between frame k and frame $k+1$, $\delta(k, 0)$ will be relatively large. The frames within a video shot are similar to each other, so $\delta(n, l)$ will also be large while $k-l \le n \le k+l$ is satisfied. Figures 1 (a) and 1 (b) show an example of this triangular pattern in the distance map.

2.2 Pattern of a Flash

If a flash occurs at frame k, the differences between frame k and frame $k-1$ and between frame k and frame $k+1$ will be relatively large while frame $k-1$ and frame $k+1$ are similar. Therefore, $\delta(k-1, 0)$ and $\delta(k, 0)$ should be large. As frames in a video shot are similar to each other, $\delta(n, l)$, $n = k-l$ or $k+1+l$, will also be large. Figures 1 (c) and 1 (d) show this two-straight-line pattern. Based on this, false alarms caused by flashes can be removed easily.

2.3 Pattern of a Gradual Transition

Detection of a gradual transition is difficult as the difference between the successive frames in the transition is relatively small. When the step between two frames in a transition is increased, their differences will be much larger than those within a video shot. From Figures 1 (e) and 1 (f), we can see that it is difficult to detect the gradual transition when l is less than 2. However, when l is larger than 6, two frames in the gradual transition will have a difference comparable to that within a cut. Compared to a cut, the pattern of a gradual transition in the distance map looks like a trapezoid instead of a triangle. This is because the consecutive frames within a neighbourhood in a gradual transition are similar.



(a) $\delta(n, l)$, $\phi(n, 4)$ and $\phi(n, 10)$ of a cut.

(b) $\delta(n, l)$, $\phi(n, 4)$ and $\phi(n, 10)$ of a flash.

(c) $\delta(n, l)$, $\phi(n, 4)$ and $\phi(n, 10)$ of a gradual transition.

Figure 2. $\delta(n, l)$, $\phi(n, 4)$ and $\phi(n, 10)$ of a cut, a flash and a gradual transition.

3 Shot Boundary Detection

Based on the analysis in Section 2, we can detect cuts and gradual transitions by analysing distance map $\delta(n, l)$.

3.1 Temporally Local Zero Mean

In order to limit the differences caused by object motion or camera motion, we remove the temporally local mean from the distance map $\delta(n, l)$. The temporally local mean of $\delta(n, l)$ is defined as follows:

$$\mu(n, L) = \frac{1}{(4L+1)\cdot L} \cdot \sum_{k=n-2L}^{n+2L} \sum_{l=0}^{L-1} \delta(k, l), \tag{2}$$

where L is the maximum step size used in $\delta(k, l)$. After being subtracted by $\mu(n, L)$, a new feature is defined as follows:

$$\eta(n, l, L) = \delta(n, l) - \mu(n, L). \tag{3}$$

3.2 Peak Detection

Based on $\eta(n, l, L)$ computed from (3), we can calculate the summation for all possible step sizes as follows:

$$\phi(n,L) = \sum_{l=0}^{L-1} \eta(n,l,L). \quad (4)$$

Figure 2 shows $\phi(n, L)$ of a cut, a flash and a gradual transition while L is equal to 4 and 9. As shown in this figure, we can observe that the pattern of shot boundaries shows like a temporally local peak in $\phi(n, L)$. For cuts, this pattern is distinct when L is larger than 3. However, for gradual transitions, the pattern is not obvious except when the step size is large enough.

Based on zero-crossing detection, frame number $k_{start}(L)$ is declared as the start point of a potential peak if $\phi(k_{start}(L)-1,L)<0$ and $\phi(k_{start}(L)+1,L)>0$. After detecting the start point, the frame number of the end point $k_{end}(L)$ is detected if $\phi(k_{end}(L)-1,L)>0$ and $\phi(k_{end}(L)+1,L)<0$. The maximum value of $\phi(n, L)$ and its frame number in each potential peak region is defined as follows:

$$\phi(k_{\max}(L,i),L) = Max(\phi(k_{start}(L,i),L),...,\phi(k,L),...,\phi(k_{end}(L,i),L)) \quad (5)$$

where L is the maximum step size, $k_{start}(L,i)$ and $k_{end}(L,i)$ denote the frame numbers of the start point and the end point of the i^{th} potential peak region, respectively, $k_{\max}(L,i)$ is the frame number where the maximum value of $\phi(n, L)$ occurs in the i^{th} potential peak region.

3.3 Cut Detection

We detect cuts by means of $k_{\max}(4,i)$. This is because the differences between the frames during a gradual transition are not distinct when the comparison step size is small, and only the cuts in $\phi(k_{\max}(4,i),4)$ have a large magnitude. Frame $k_{\max}(4,i)$ is declared as a cut transition if $\phi(k_{\max}(4,i),4)>\theta_{cut}$ and $\eta(k_{\max}(4,i),0,4)\geq 2$. We also apply a post-processing step to remove false alarms caused by flashes. As illustrated in Figure 1, the pattern of a flash looks like a hollow triangle. Therefore, $k_{\max}(4,i)$ remains as a cut only if $0.5<\eta(k_{\max}(4,i),0,4)/\eta(k_{\max}(4,i),1,4)<2$.

3.4 Gradual Transition Detection

Gradual transitions are detected using $k_{\max}(10,i)$. Like cut detection, frames from $k_{start}(10,i)$ to $k_{end}(10,i)$ are declared as a gradual transition if $\phi(k_{\max}(10,i),10)>\theta_{gradual}$ and $\eta(k_{\max}(10,i),0,10)<2$. As the maximum comparative step size used in our proposed algorithm is 10, a long gradual transition is divided into several consecutive gradual transitions. To overcome this drawback, we merge successive gradual transitions into a single one if the distance between them is less than 10 frames.

4 Experimental Results

Our algorithm was developed for the TRECVID (Petersohn 2004, TREC 2005) 2005 SBD testing set as shown in Table 1. The parameters θ_{cut} and $\theta_{gradual}$ used in ten runs are listed in Table 2. Tables 3 and 4 and Figures 3, 4, 5, and 6 show the evaluation results. Cut detection is more accurate and robust than gradual transition detection. From Table 3, we can see that when θ_{cut} = 5 and $\theta_{gradualt}$ = 9.5, our proposed algorithm achieves the best performance. The recall and precision rates for gradual transition detection, as shown in Table 4, are around 70%, except sequence ***20041108_120000_NTDTV_NTDNEWS12_CHN*** in the testing set. The reason for this is that this sequence contains several parts of news captured by hand, so it is full of fast camera motions such as pans, tilts and zooms.

In this evaluation, we found a problem caused by the video decoder used. At the beginning, we evaluated our system using the Microsoft MPEG-1 Video Decoder – Version 2.0. From the evaluation results, the performances of our system for the three videos with sequence numbers 9, 10, and 11 are very poor. From the file: http://www-nlpir.nist.gov/projects/tvpubs/tvpapers04/tv4overview.pdf, we found that VirtualDub(Lee, 2001) was recommended for viewing the videos and the frame numbers. Compared to VirtualDub1.5.10, we found that, e.g. the sequence ***NASADT18***, when the frame number is higher than 20000, the frame numbers decoded by Microsoft MPEG-1 Video Decoder are 190 to 292 smaller than those decoded by VirtualDub. Nevertheless, for the sequence ***20041115_133000_MSNBC_MSNBCNEWS13_ENG***, the frame numbers decoded by the two decoders are the same. Therefore, in this paper, we present the evaluation results for our algorithm using the VirtualDub video decoder rather than the Microsoft decoder.

Table 1. The names of the video sequence in the TRECVID 2005 evaluation set.

Sequence Number	Sequence Name
1	20041102_160001_CCTV4_DAILY_NEWS_CHN
2	20041106_110000_MSNBC_MSNBCNEWS11_ENG
3	20041108_120000_NTDTV_NTDNEWS12_CHN
4	20041115_133000_MSNBC_MSNBCNEWS13_ENG
5	20041116_120100_NTDTV_NTDNEWS12_CHN
6	20041118_183000_NBC_NIGHTLYNEWS_ENG
7	20041118_230000_NBC_NBCPHILA23_ENG
8	20041119_140000_LBC_LBCNAHAR_ARB
9	NASAConnect-AO
10	NASAConnect-HiddenTreasures
11	NASADT18
12	NASASF-TheTechnicalKnockout

Table 2. Parameters of the submissions.

Run	1	2	3	4	5	6	7	8	9	10
θ_{cut}	**10**	**10**	**10**	**9**	**10**	**10.8**	**9**	**5**	**3**	**12**
$\theta_{gradual}$	**3.5**	**4.5**	**5.5**	**6.5**	**7.5**	**8.5**	**8.75**	**9.5**	**10.5**	**11.5**

Table 3. Evaluation results of the ten runs.

Run	Cut				Gradual			
	Recall		Precision		Recall		Precision	
	$N_{correct}/(N_{correct}+N_{missed})$		$N_{correct}/(N_{correct}+N_{missed})$		$N_{correct}/(N_{correct}+N_{missed})$		$N_{correct}/(N_{correct}+N_{missed})$	
1	**3059/3380**	**90.50**	**3059/3287**	**93.06**	**1011/1155**	**87.53**	**1011/2291**	**44.13**
2	**3059/3380**	**90.50**	**3059/3283**	**93.18**	**990/1155**	**85.71**	**990/1945**	**50.90**
3	**3061/3380**	**90.56**	**3061/3285**	**93.18**	**977/1155**	**84.59**	**977/1724**	**56.67**
4	**3078/3380**	**91.07**	**3078/3310**	**93.00**	**952/1155**	**82.42**	**952/1554**	**61.26**
5	**3061/3380**	**90.56**	**3061/3280**	**93.32**	**931/1155**	**80.61**	**931/1439**	**64.70**
6	**3047/3380**	**90.15**	**3047/3252**	**93.70**	**906/1155**	**78.44**	**906/1344**	**67.41**
7	**3078/3380**	**91.07**	**3078/3310**	**92.99**	**901/1155**	**78.01**	**901/1326**	**67.95**
8	**3112/3380**	**92.07**	**3112/3419**	**91.02**	**874/1155**	**75.67**	**874/1248**	**70.03**
9	**3119/3380**	**92.28**	**3119/3506**	**88.96**	**829/1155**	**71.77**	**829/1153**	**71.90**
10	**3027/3380**	**89.56**	**3027/3227**	**93.80**	**793/1155**	**68.66**	**793/1084**	**73.15**

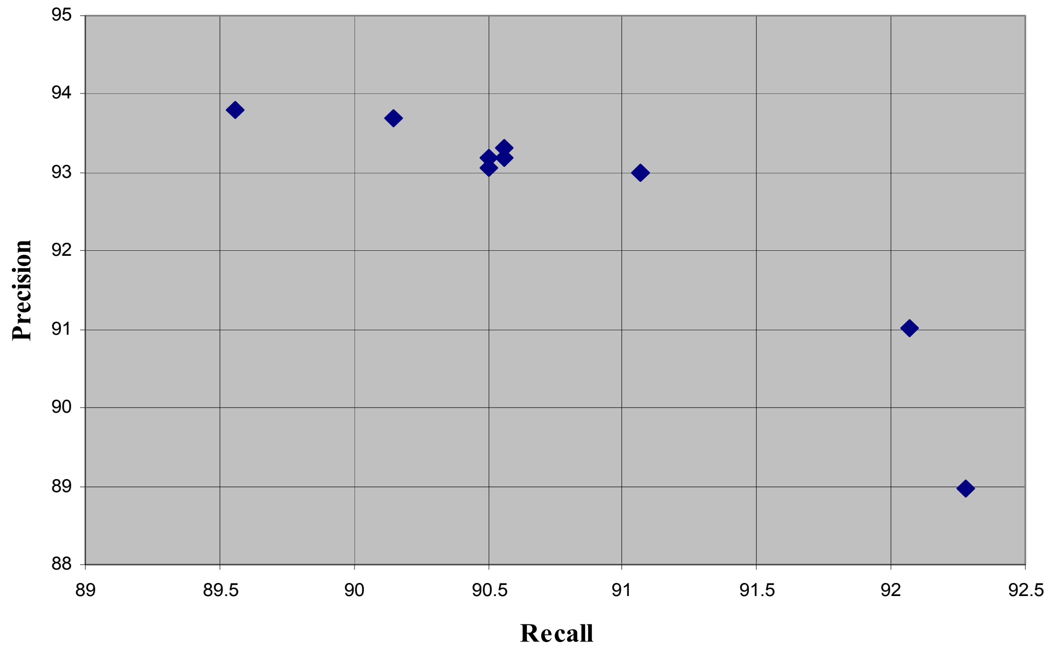


Figure 3. Recall and precision rates of cut detection for the ten runs.

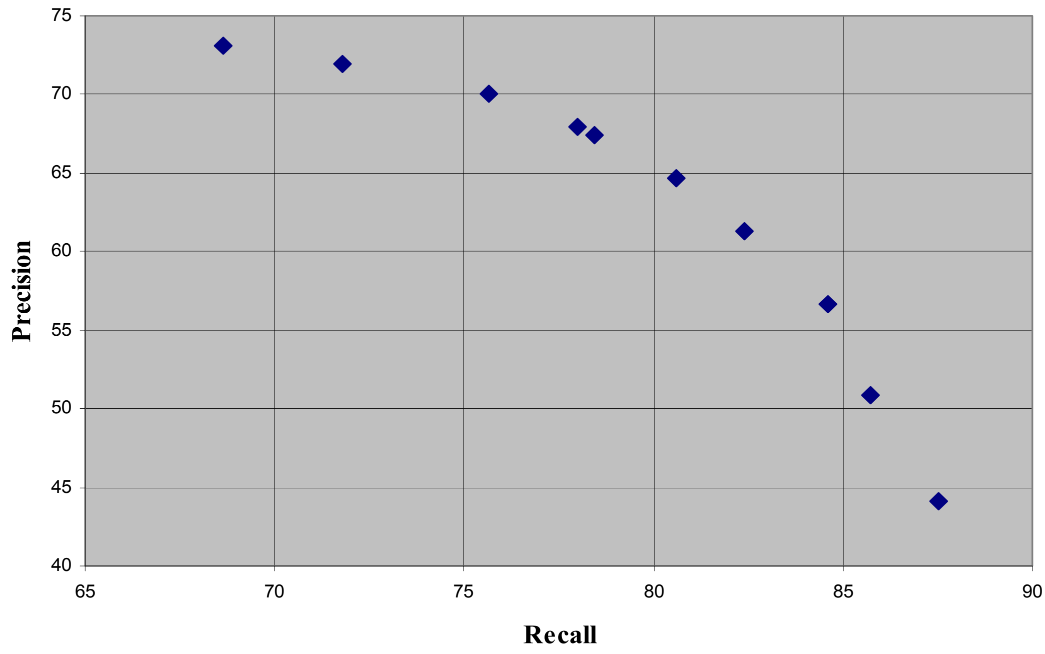


Figure 4. Recall and precision rates of gradual transition detection for the ten runs.

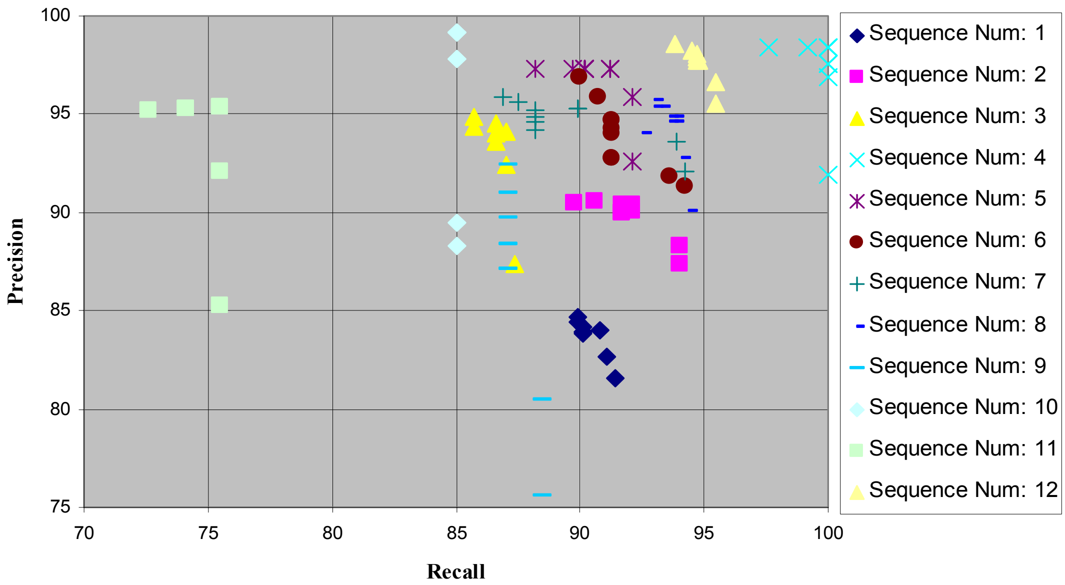


Figure 5. Recall and precision rates for cut detection of each sequence.

Table 4. Evaluation results of each sequence of the ten runs.

Sequence Number		Run	1	2	3	4	5	6	7	8	9	10
1	Cut	Recall	**90.1**	**90.1**	**90.1**	**90.8**	**90.1**	**89.9**	**90.8**	**91.1**	**91.4**	**89.9**
		Precision	**83.9**	**84.2**	**83.8**	**84.0**	**83.9**	**84.4**	**84.0**	**82.7**	**81.6**	**84.7**
	Gradual	Recall	**90.9**	**89.1**	**86.4**	**81.9**	**78.3**	**70.2**	**69.3**	**66.6**	**59.4**	**54.0**
		Precision	**40.8**	**46.2**	**49.4**	**52.9**	**55.0**	**55.7**	**56.2**	**60.6**	**64.0**	**62.5**
2	Cut	Recall	**91.7**	**91.7**	**91.7**	**92.1**	**91.7**	**90.6**	**92.1**	**94.0**	**94.0**	**89.8**
		Precision	**90.0**	**90.0**	**90.0**	**90.1**	**90.4**	**90.6**	**90.4**	**88.3**	**87.4**	**90.5**
	Gradual	Recall	**78.7**	**77.6**	**77.6**	**77.6**	**77.6**	**75.5**	**75.5**	**75.5**	**74.4**	**72.3**
		Precision	**54.0**	**59.3**	**64.0**	**67.5**	**70.8**	**71.7**	**71.7**	**73.1**	**73.6**	**74.7**
3	Cut	Recall	**86.6**	**86.6**	**86.6**	**87.0**	**86.6**	**85.7**	**87.0**	**87.0**	**87.4**	**85.7**
		Precision	**93.6**	**93.6**	**94.0**	**94.1**	**94.5**	**94.4**	**94.1**	**92.4**	**87.4**	**94.9**
	Gradual	Recall	**79.5**	**75.5**	**69.3**	**61.2**	**57.1**	**57.1**	**57.1**	**55.1**	**53.0**	**51.0**
		Precision	**30.7**	**33.9**	**35.4**	**35.2**	**35.4**	**37.8**	**37.8**	**37.5**	**39.3**	**40.3**
4	Cut	Recall	**100**	**100**	**100**	**100**	**100**	**99.2**	**100**	**100**	**100**	**97.6**
		Precision	**98.4**	**98.4**	**97.6**	**97.6**	**98.4**	**98.4**	**98.4**	**96.9**	**91.9**	**98.4**
	Gradual	Recall	**86.6**	**84.4**	**84.4**	**82.2**	**80.0**	**80.0**	**80.0**	**75.5**	**75.5**	**73.3**
		Precision	**45.3**	**50.6**	**58.4**	**60.6**	**66.6**	**72.0**	**73.4**	**72.3**	**72.3**	**80.4**
5	Cut	Recall	**90.2**	**90.2**	**90.2**	**91.2**	**90.2**	**89.7**	**91.2**	**92.1**	**92.1**	**88.2**
		Precision	**97.3**	**97.3**	**97.3**	**97.3**	**97.3**	**97.3**	**97.3**	**95.9**	**92.6**	**97.3**
	Gradual	Recall	**91.4**	**88.5**	**85.7**	**85.7**	**85.7**	**82.8**	**82.8**	**80.0**	**80.0**	**71.4**
		Precision	**31.3**	**36.0**	**37.5**	**42.8**	**45.4**	**45.3**	**46.7**	**50.0**	**52.8**	**55.5**
6	Cut	Recall	**91.3**	**91.3**	**91.3**	**91.3**	**91.3**	**90.7**	**91.3**	**93.6**	**94.2**	**90.0**
		Precision	**94.3**	**94.0**	**94.7**	**92.8**	**94.7**	**95.9**	**92.8**	**91.8**	**91.3**	**96.9**
	Gradual	Recall	**92.5**	**92.5**	**90.6**	**88.1**	**86.8**	**86.8**	**86.8**	**84.3**	**80.0**	**77.5**
		Precision	**52.8**	**58.7**	**63.3**	**65.8**	**71.2**	**75.9**	**75.9**	**78.9**	**81.5**	**83.7**
7	Cut	Recall	**88.2**	**88.2**	**88.2**	**89.9**	**88.2**	**87.5**	**89.9**	**93.9**	**94.2**	**86.9**
		Precision	**94.2**	**94.6**	**94.9**	**95.3**	**95.2**	**95.6**	**95.3**	**93.6**	**92.1**	**95.9**
	Gradual	Recall	**83.9**	**82.1**	**81.2**	**78.5**	**75.0**	**74.1**	**74.1**	**72.3**	**67.8**	**66.0**
		Precision	**35.7**	**44.0**	**53.2**	**61.5**	**65.1**	**69.7**	**70.3**	**72.3**	**73.7**	**74.7**
8	Cut	Recall	**93.0**	**93.0**	**93.6**	**93.8**	**93.6**	**93.3**	**93.8**	**94.1**	**94.4**	**92.5**
		Precision	**95.4**	**95.7**	**94.6**	**94.9**	**94.9**	**95.4**	**94.6**	**92.8**	**90.1**	**94.0**
	Gradual	Recall	**72.8**	**69.9**	**69.9**	**68.9**	**66.9**	**66.9**	**66.0**	**65.0**	**60.1**	**57.2**
		Precision	**32.8**	**38.5**	**47.6**	**58.6**	**66.3**	**68.3**	**68.6**	**69.7**	**72.0**	**73.7**
9	Cut	Recall	**87.1**	**87.1**	**87.1**	**87.1**	**87.1**	**87.1**	**87.1**	**88.5**	**88.5**	**87.1**
		Precision	**87.1**	**87.1**	**88.4**	**89.7**	**91.0**	**91.0**	**88.4**	**80.5**	**75.6**	**92.4**
	Gradual	Recall	**94.0**	**94.0**	**93.4**	**92.2**	**91.6**	**91.6**	**91.0**	**89.8**	**86.3**	**84.5**
		Precision	**64.4**	**71.4**	**75.4**	**79.4**	**81.4**	**84.1**	**84.5**	**86.2**	**87.8**	**88.7**
10	Cut	Recall	**85.0**	**85.0**	**85.0**	**85.0**	**85.0**	**85.0**	**85.0**	**85.0**	**85.0**	**85.0**
		Precision	**99.2**	**99.2**	**99.2**	**97.8**	**99.2**	**99.2**	**97.8**	**89.5**	**88.3**	**99.2**
	Gradual	Recall	**94.5**	**93.2**	**93.2**	**90.5**	**90.5**	**86.4**	**86.4**	**83.7**	**78.3**	**74.3**
		Precision	**41.9**	**47.5**	**53.0**	**58.2**	**62.0**	**64.0**	**65.9**	**68.1**	**69.0**	**69.6**
11	Cut	Recall	**74.1**	**74.1**	**74.1**	**75.5**	**74.1**	**74.1**	**75.5**	**75.5**	**75.5**	**72.6**
		Precision	**95.3**	**95.3**	**95.3**	**95.4**	**95.3**	**95.3**	**95.4**	**92.1**	**85.3**	**95.2**
	Gradual	Recall	**87.1**	**81.6**	**81.6**	**81.6**	**79.8**	**77.9**	**76.1**	**74.3**	**70.6**	**68.8**
		Precision	**57.9**	**64.4**	**68.9**	**70.6**	**72.5**	**73.9**	**73.4**	**73.6**	**75.4**	**75.7**
12	Cut	Recall	**94.7**	**94.7**	**94.7**	**94.8**	**94.7**	**94.5**	**94.8**	**95.5**	**95.5**	**93.8**
		Precision	**97.7**	**97.9**	**98.1**	**97.7**	**97.9**	**98.2**	**97.7**	**96.6**	**95.5**	**98.6**
	Gradual	Recall	**90.5**	**88.4**	**87.3**	**84.2**	**81.0**	**73.6**	**73.6**	**66.3**	**62.1**	**55.7**
		Precision	**35.1**	**45.1**	**52.8**	**55.5**	**57.4**	**60.3**	**61.4**	**63.6**	**64.1**	**63.0**

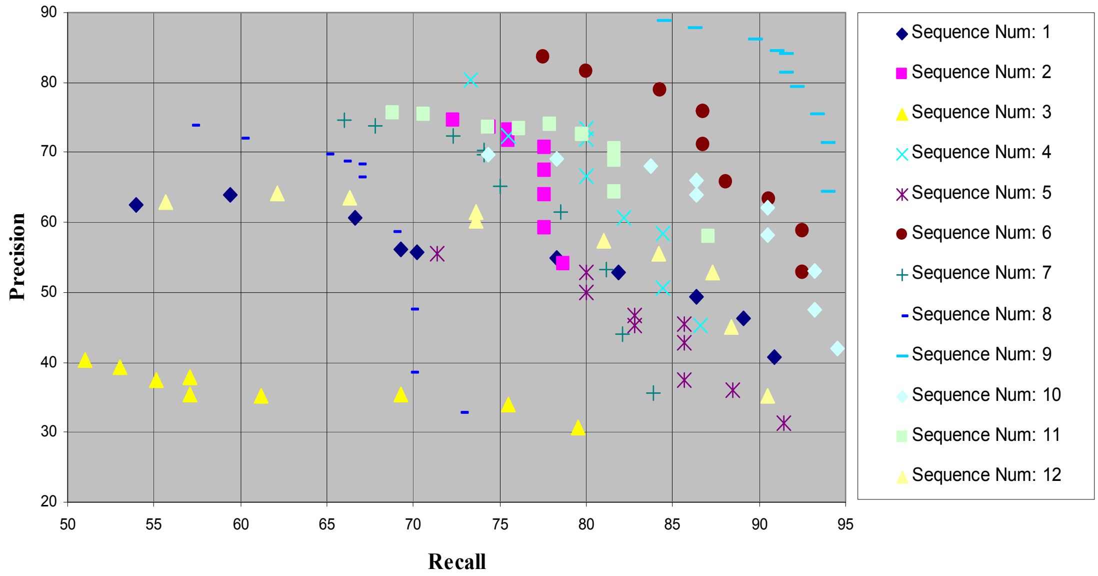


Figure 6. Recall and precision rates for gradual transition detection of each sequence.

5 Conclusion

In this paper, we have proposed a scheme which converts a 3D video sequence into a 2D distance map based on multi-step comparison for shot boundary detection. In the distance map, a cut appears like a triangle, a flash forms two straight lines, and a gradual transition results in a trapezoid. Cut detection is more accurate as the difference between two successive frames is relatively large. When the contents between two shots change gradually, the difference between consecutive frames is small. Therefore, comparison between frames at a greater distance apart is needed. In our system, gradual transition is detected by analysing the differences between frames using multiple step sizes. Our algorithm can handle complex transitions, but it uses the metric based on the colour histogram only and has not considered motions. In addition, two different shots may have similar histograms. Thus, motions might be wrongly classified as gradual transitions. To solve this problem, more sophisticated metric and motion estimation will be investigated and employed in our future research.

6 References

Lienhart, R. (1999): Comparison of Automatic Shot Boundary Detection Algorithms. *Proc. IS&T/SPIE on Storage and Retrieval for Image and Video Databases VII*, San Jose, CA, USA, **3656**:290-301.

Lienhart, R. (2001): Reliable Dissolve Detection. *Proc. SPIE Storage and Retrieval for Media Databases*, **4315**:219-230.

Yeo, B.L., and Liu, B. (1995): Rapid Scene Analysis on Compressed Video. *Journal of IEEE Transactions on Circuits and Systems for Video Technology* **5**(6):533-544.

Zhang, H.J., Kankanhalli, A., and Smoliar, S.W. (1993): Automatic Partitioning of Full-Motion Video. *Journal of Multimedia Systems* **1**:10-28.

Zabih, R., Miller, J., and Mai, K. (1999): A Feature-Based Algorithm for Detecting and Classifying Production Effects. *Journal of Multimedia Systems* **7**:119-128.

Bouthemy, P. Gelgon, M., and Ganansia, F. (1999): A Unified Approach to Shot Change Detection and Camera Motion Characterization. *Journal of IEEE Transactions on Circuits and Systems for Video Technology* **9**(7):1030-1044.

Pei, S. C., and Chou, Y. Z. (1999): Efficient MPEG Compressed Video Analysis Using Macroblock Type Information. *Journal of IEEE Transactions on Multimedia* **1**:321-333.

Jun, S.B., Yoon, K., and Lee, H.Y. (2000): Dissolve Transition Detection Algorithm Using Spatio-Temporal Distribution of MEPG Macro-Block Types. *Proc.* 8th *ACM Multimedia,* Marina del Rey, California, USA, 391-394.

Alattar, A.M. (1993): Detecting and Compressing Dissolve Regions in Video Sequences with a DVI Multimedia Image Compression Algorithm. *Proc. IEEE International Symposium on Circuit and Systems*, **1**:13-16.

Ngo, C.W., Pong, T.C., and Chin, R.T. (2001): Video Partitioning by Temporal Slice Coherency. *Journal of IEEE Transactions on Circuits and Systems for Video Technology*, **11**(8):941-953.

Han, S.H., and Kweon, I.S. (2003): Detecting Cuts and Dissolves Through Linear Regression Analysis. *Journal of Electronics Letters*, **39**:1579-1581.

Ngo, C.W. (2003): A Robust Dissolve Detector by Support Vector Machine. *Proc.* 11th *ACM International Conference on Multimedia*, Berkeley, CA, USA, 283-286.

Hanjalic, A. (2002): Shot Boundary Detection: Unraveled and Resolved? *Journal of IEEE Transactions on Circuits and Systems for Video Technology*, **12**(2):90-105.

Vasconcelos, N., and Lippman, A. (2000): Statistical Models of Video Structure for Content Analysis and Characterization. *Journal of IEEE Transactions on Image Processing*, **9**(1):3-19.

Liu, T.Y., Lo, K.T., Zhang, X.D., and Feng, J. (2004): A New Cut Detection Algorithm with Constant False-Alarm Ratio for Video Segmentation. *Journal of Visual Communication and Image Representation*, **15**(2):132-144.

Cai, C., Lam, K.M., and Tan, Z. (2004): An Efficient Scene Break Detection Based on Linear Prediction. *Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, 555-558.

Cai, C., Lam, K.M., and Tan, Z. (2005): A Unified Shot Boundary Detection Method Based on Linear Prediction with Bayesian Cost Functions. *Proc. IEEE International Workshop on VLSI Design and Video Technology*, Suzhou, China, 101-104.

Bescós, J. (2004): Real-Time Shot Change Detection Over Online MPEG-2 Video. *Journal of IEEE Transactions on Circuits and Systems for Video Technology* **14**(4):475-484.

Petersohn, C. (2004): "Fraunhofer HHI at TRECVID 2004: Shot Boundary DetectionSystem", TREC Video Retrieval Evaluation Online Proceedings.

TREC Video Retrieval Evaluation: Digital Video Retrieval at NIST. http://www-nlpir.nist.gov/projects/trecvid/.