World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:6, No:7, 2012

Fast Algorithm of Shot Cut Detection

Lenka Krulikovská, Jaroslav Polec, Tomáš Hirner

***Abstract*—**In this paper we present a novel method, which reduces the computational complexity of abrupt cut detection. We have proposed fast algorithm, where the similarity of frames within defined step is evaluated instead of comparing successive frames. Based on the results of simulation on large video collection, the proposed fast algorithm is able to achieve 80% reduction of needed frames comparisons compared to actually used methods without the shot cut detection accuracy degradation.

***Keywords*—**Abrupt cut, fast algorithm, shot cut detection, Pearson correlation coefficient.

I. Introduction

PROGRES in the multimedia compression technology and computer performance has led to the widespread availability of digital video. There is a corresponding growth in the need for methods to reliably detect shot boundaries within the video sequence.

The detection of shot boundaries provides a base for nearly all video abstraction and high-level video segmentation approaches, like for example searching, browsing, indexing and fast forwarding.

Therefore, solving the problem of shot-boundary detection is one of the major prerequisites for revealing higher level video content structure. Moreover, other research areas can profit considerably from successful automation of shot-boundary detection processes as well.

There are a number of different types of transitions or boundaries between shots, like an abrupt cut, a fade in/out, a dissolve and a wipe. Of course, many other types of gradual transition are possible [1].

A cut is an abrupt shot change that occurs in a single frame, an example is shown in Fig. 1. A fade is a slow change in brightness usually resulting in or starting with a solid black frame. A dissolve occurs when the images of the first shot get dimmer and the images of the second shot get brighter, with frames within the transition showing one image superimposed on the other. A wipe occurs when pixels from the second shot replace those of the first shot in a regular pattern such as in a line from the left edge of the frames.

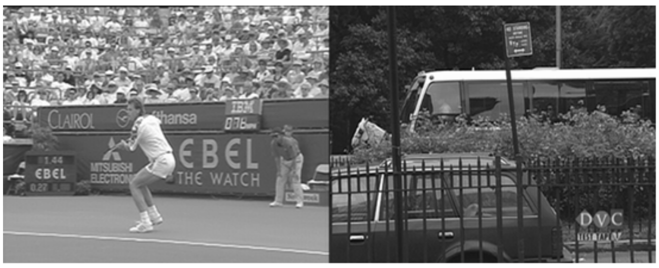


Fig. 1 An example of abrupt cut

T. Hirner is with the Institute of Telecommunications, University of Technology, Ilkovičova 3, 812 19, Bratislava, Slovak Republic (phone: +421903248068, email: tomas.hirner@gmail.com).

In general, abrupt transitions are much more common than gradual transitions, accounting for over 99% of all transitions found in video [2]. Therefore, the correct detection of abrupt shot transitions is a very important task in video segmentation and this paper is focused only on the detection of an abrupt cut. Different approaches have been proposed to extract shots. The major techniques used for the shot boundary detection are pixel differences, statistical differences, histogram comparisons [3], edge differences, compression differences and motion vectors [4]-[6].

There are various possibilities for improving on the basic methods. The variety of basic methods opens up the possibility of combining several of them into a multiple expert framework, explored in [7]-[9].

Also, one can use an adaptive threshold setting, by using statistics of the dissimilarity measure within a sliding window [10]-[12]. The published methods apply frame by frame comparison, it means the pairs of successive frames are compared and their similarity is evaluated by selected measure. This approach leads to high computational complexity due to number of needed frames comparisons, where each frame is represented by huge matrix (the size of matrix depends on video resolution) for each color component. If we have video sequence with *n* frames, *n-1* frames comparisons are required by actually used shot cut detection algorithms.

Our aim was to propose novel fast algorithm, which would reduce the number of required comparisons and speed-up the process of shot cut detection without degrading the accuracy.

The paper is structured as follows: in the second section the proposed fast algorithm is described. The achieved results are in the third section. All results are summarized and discussed in conclusion.

II. Proposed Fast Algorithm

The majority of shot cut detection methods use frame by frame comparison. Two successive frames in the video sequence are compared and they are classified as cut or non-cut by used threshold.

In our approach the frames are also compared and the cut position is determined according to their similarity evaluated by selected measure and threshold. The difference and the novelty of presented method compared to existing ones is in the selection of frames to be compared and in the way how the shot change is consequently found`.

The frame set as actual is compared to the frame distant by a defined step. We decide if the frames are within one shot or not based on their similarity. If frames are within one shot, the distant frame is set as new actual frame. If the compared frames belong to different shots, the procedure for searching the position of shot change starts.

At first the step is divided by 2. Then the actual frame is compared to the frame distant by reduced step. Again, it is needed to decide if the frames are within one shot based on the threshold and the step is reduced by 2.

If compared frames belong to one shot, the actual frame is compared to the frame distant by previous distance plus new step (as we are still in one shot we have to move forward to find the shot change).

In the second case, the actual frame is compared to the frame distant by previous distance minus new step (as we are actually in the next shot, we have to move back to find the shot change). These steps are repeated while the reduced step is equal to 1 and the found frame is saved as a potential candidate for cut and marked as actual frame.

After evaluating whole video sequence we have a set of candidates for shot changes. There would be a lot of false detections among them due to video content variations within shots, therefore we perform a frame by frame comparison for each frame in the set. The range for frame by frame comparison for frame *i* is *<i-2;i+2>*.

If the candidate is confirmed as cut, it is saved to final set of cuts, otherwise it is dropped as false detection. Fig. 2 shows an illustration example for better understanding the principle of fast algorithm.

The fast algorithm has few limitations: it cannot be used in the real time applications; the defined step has to be the power of two; we have to use the similarity measure with the known range of values for appropriate threshold determination, because an adaptive threshold cannot be used.

Therefore we have employed the absolute value of Pearson correlation coefficient as metric for evaluating the similarity of compared frames.

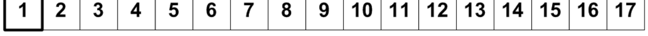
In statistics, the Pearson's correlation coefficient has been widely employed to measure the correlation (or strength of linear dependence) between two variables X and Y [13]. The value for a Pearson correlation coefficient can fall between -1 and 1, where 0 means no correlation. Generally, correlations above 0.80 are considered as really high.

Pearson correlation coefficient is calculated as:

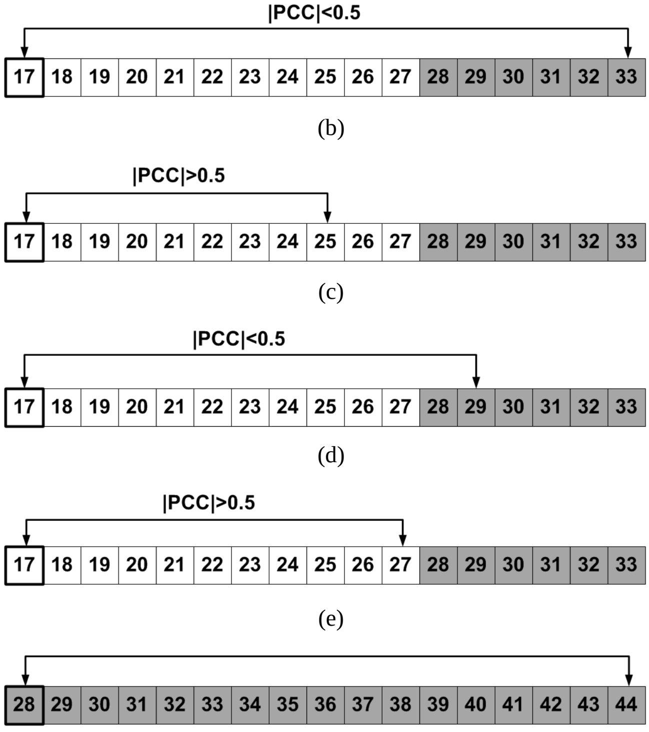
$$PCC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(X(i,j)-X^m)(Y(i,j)-Y^m)}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(X(i,j)-X^m)^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(Y(i,j)-Y^m)^2}} \quad (1)$$

, where X^m and Y^m stand for mean pixel intensity of frames X and Y.

According to the definition the absolute value of Pearson correlation coefficient achieved the values between 0 and 1. We have set fixed threshold to 0.5. All values below threshold are classified as shot cut.



(a)



(f)

Fig. 2 An illustration example of the principle of fast algorithm

Fig. 2 shows an example how the proposed fast algorithm works. The defined step is equal to 16, squares represent frames, shots differ by the color of square fill, the actual frame is marked by border weight and the arrow indicates which frames are compared.

At first (Fig. 2 a)) the actual frame (frame 1) is compared to the frame distant by a step (frame 17). Pearson correlation coefficient value is above the defined threshold thus we are still within one shot. Frame 17 becomes new actual frame.

Then (Fig. 2 b)) the new actual frame (frame 17) is compared to frame 33. According to classification by threshold these frames belongs to different shots. Therefore the step is reduced by two and we have to move back to find the place of shot change.

Actual frame is compared to the frame 25, as it is displayed in Fig. 2 c). These frames belong to one shot, so we have to reduce step and move forward to find a cut.

In the next step (Fig. 2 d)) the actual frame is compared to frame 29. Based on the evaluation metric value the frame 29 belongs to the next shot and we have to go back with reduced step to locate cut.

In the Fig. 2 e) the actual frame is compared to frame 27. We can see the frames 17 and 27 are within one shot and we should move forward by reduced step. But, the step after reducing equals 1, therefore no more frames are compared and we save frame 28 as candidate for a cut and set it as new actual frame. This procedure is applied to whole video sequence as Fig. 2 f) indicates where new actual frame is compared with distant frame.

III. Experimental Results

The proposed fast algorithm and its accuracy were evaluated through test experiment. The detection technique was applied to several TRECVID video test sets that exhibit different types of shots and contain significant object and camera motion inside the shots.

We have used video sequences at CIF resolution (352 x 288 pixels) with 420 abrupt cuts. The accuracy of shot detection and the reduction of needed frames comparisons were evaluated for the step values to 4, 8, 16, 32, 64, 128 and 256. As similarity metric Pearson correlation coefficient was used and the threshold value was set to 0.5.

For the evaluation of shot boundary detection accuracy we have used standard metrics recall (r), precision (p) and F1 score (F1) [1].

Recall is the fraction of all known transitions that are correctly detected, while precision is the fraction of reported transitions that match the known transitions recorded in the reference data. F1 score takes into account both missed cuts and false detections thus it gives global overview of shot transition detection algorithm accuracy.

In all tables C stands for correctly detected cuts, M for missed cuts, F for false detected cuts, p for precision measure, r for recall measure and F1 for F1 score measure.

Table I shows the evaluation of accuracy of the fast algorithm. The best results according to correctly detected and missed cuts were achieved for first three step values 4, 8 and 16. The lowest number of false detection was reached by step equals to 32, 128 and 256.

For precision measure the accuracy higher than 99% was achieved by all simulated steps values. The highest accuracy, more than 99.5%, was reached for steps 8 and 16.

The values of recall are in the range from 73% to 98%, where the highest accuracy was obtained for steps 4, 8, 16 and the lowest for step value 256.

TABLE I
THE ACCURACY OF FAST ALGORITHM

Step	C	M	F	p	r	F1
4	412	8	4	0.99038	0.98095	0.98565
8	412	8	2	0.99517	0.98095	0.98801
16	412	8	2	0.99517	0.98095	0.98801
32	404	16	1	0.99753	0.9619	0.97939
64	395	25	2	0.99496	0.94048	0.96695
128	360	60	1	0.99723	0.85714	0.9219
256	308	112	1	0.99676	0.73333	0.84499

As mentioned above, F1 score gives more global overview of shot cut detection accuracy than precision and recall. From this perspective the best result was achieved for step 8 and 16 and the worst one for step 256.

However accuracy for all steps reached pretty high values, starting at more than 84.4%.

TABLE II
THE COMPLEXITY REDUCTION BY FAST ALGORITHM

Step	Fast algorithm	Frame by frame approach	Comparisons reduction [%]
4	27888	93631	70.215
8	20994	93631	77.578
16	20066	93631	78.569
32	20718	93631	77.873
64	21892	93631	76.619
128	21656	93631	76.871
256	21867	93631	76.646

Table II illustrates the ability of fast algorithm to reduce computational complexity. Simulated values of step are listed in the first column. The second column represents the number of needed comparison for fast algorithm. This number includes all comparison performed during shot boundary detection process including final search procedure. Third column shows the number of comparisons required by frame by frame comparison. The fourth column represents the reduction of frames comparison achieved by fast algorithm compared to frame by frame approach.

Fast algorithm was able to achieve the reduction from 70% to 79%. The highest reduction (78.56%) was reached for step value equal to 16 and the lowest one (70.21%) for step 4. The rest of simulated steps gives similar results (between 76.5% and 77%), except steps 8 and 32, where the reduction achieved value more than 77,5%.

Table III gives global overview on achieved accuracy and comparison reduction of fast algorithm. Based on the results, the step values 8, 16, 32, 64 and 128 achieved high accuracy and reduction. For step 4 there is lower reduction, because fast algorithm required more comparisons during shot cut detection. Step 256 reached the worst accuracy due to larger amount of missed detection as we compared very distant frames. The best result in both accuracy and reduction was achieved with step 16, where the accuracy of shot detection is 98.8% and reduction is 78.57%.

TABLE III
THE ACCURACY AND REDUCTION ACHIEVED BY FAST ALGORITHM

Step	C	M	F	F1	Reduction by fast algorithm [%}
4	412	8	4	0.98565	70.215
8	412	8	2	0.98801	77.578
16	412	8	2	0.98801	78.569
32	404	16	1	0.97939	77.873
64	395	25	2	0.96695	76.619
128	360	60	1	0.9219	76.871
256	308	112	1	0.84499	76.646

IV. Conclusion

In this paper we presented a novel fast algorithm for abrupt cut detection. Actually used methods employ frame by frame comparison, where pairs of successive frames are compared.

Therefore these methods are highly time demanding and computationally complex. Proposed fast algorithm solves mentioned limitation for applications, which do not need perform in real time.

In the proposed fast algorithm the frame set as actual is compared to frame distant by a defined step. Based on the similarity measure result and defined threshold we determine if compared frames belong to the same shot or to different shots. Subsequently procedure for each case is employed.

The proposed algorithm has few limitations: the step has to be the power of two and it is not possible to use adaptive threshold therefore a measure with known range of values has to be used for the evaluation of similarity of compared frames.

We have evaluated the proposed algorithm through test experiment on large dataset. We have simulated step values 4, 8, 16, 32, 64, 128 and 256. Pearson correlation coefficient was used as similarity metric and the threshold was set to 0.5.

All steps, except the smallest and the largest one, achieved pretty high accuracy and reduction of required comparison during shot cut detection. The values of accuracy for F1 score measure are in the range from 92% to 99%. The reduction of needed frames comparison is from 76% to 79%.

The smallest step 4 achieved worse reduction due to larger number of performed comparisons. For the largest step 256 the accuracy was lower, because a lot of missed detections caused by comparing very distant frames.

The best result in both accuracy and comparisons reduction was achieved with step 16, where the accuracy of shot detection is 98.8% and reduction is 78.57%.

We can state that proposed fast algorithm gives reliable results for shot boundary detection and is able to reduce computational and time complexity of actually used methods.

Acknowledgement

Research described in the paper was financially supported by the Slovak Research Grant Agency (VEGA) under grant No. 1/0602/11.

References

[1] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization", IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no.1, page(s): 82-91, January 2006

[2] S. PASCHALAKIS and D. SIMMONS, (2008, April 24), “Detection of gradual transitions in video sequences” [Online]. Available: http://www.wipo.int/pctdb/en/wo.jsp?WO=2008046748&IA=EP2007060594&DISPLAY=STATUS.

[3] A. Amiri and M. Fathy, “Video shot boundary detection using QR decomposition and Gaussian transition detection,“ EURASIP Journal on Advances in Signal Processing, Volume 2009, Article ID 509438.

[4] A. Hanjalic, “Shot-boundary detection: unraveled and resolved?,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 2, pp. 90–105, 2002.

[5] J. S. Boreczky and L. A. Rowe, “Comparison of video shot boundary detection techniques,” Storage and Retrieval for Still Image and Video Databases IV, Proc. SPIE 2664, pp. 170-179, 1996.

[6] R. Lienhart, “Comparison of automatic shot boundary detection algorithms,” Storage and Retrieval for Image and Video Databases VII, vol. 3656 of Proceedings of SPIE, pp. 290–301, San Jose, Ca, USA, 1999.

[7] M. R. Naphade, R. Mehrotra, A. M. Ferman, J. Warnick, T. S. Huang and A. M. Tekalp, “A high-performance shot boundary detection algorithm using multiple cues,” Proc. IEEE Int. Conf. on Image Proc., volume 2, pages 884–887, 1998.

[8] C. Taskiran and E. J. Delp, “Video scene change detection using the generalized sequence trace,” Proc. IEEE Int. Conf. on Image Proc., pages 2961–2964, 1998.

[9] Y. Yusoff, J. Kittler and W. Christmas, “Combining multiple experts for classifying shot changes in video sequences,” Proc. 6th Int. Conf. on Multimedia Comp. and Systems (ICMCS), volume 2, pages 700–704, Florence, Italy, 1999.

[10] R. Dugad, K. Ratakonda and N. Ahuja, “Robust video shot change detection”, IEEE Workshop on Multimedia Signal Processing, 1998.

[11] B. L. Yeo and B. Liu, “Rapid scene analysis on compressed video,” IEEE Trans. On Circuits and Systems for Video Technology, 5(6):533–544, 1995.

[12] R. Zabih, J. Miller and K. Mai, “A feature-based algorithm for detecting and classifying production effects,” ACM Multimedia Systems, 7(2):119–128, 1999.

[13] Y. K. Eugene, and R. G. Johnston, “The Ineffectiveness of the Correlation Coefficient for Image Comparisons,” Technical Report LA-UR-96-2474, Los Alamos, 1996.

[14] M. Oravec, J. Pavlovičová, J. Mazanec, Ľ. Omelina, M. Féder and J. Ban, "Efficiency of Recognition Methods for Single Sample per Person Based Face Recognition," *Reviews, Refinements and New Ideas in Face Recognition (InTech)*, pp. 181-206, 2011.