FAST SHOT DETECTION FOR HIGH QUALITY LOW DELAY H.264 VIDEO CODING

P. Usach, J. Sastre, V. Naranjo[1] and J.M. López†*

Instituto de Tecnologías y Aplicaciones Multimedia (iTEAM)
Universidad Politécnica de Valencia. Camino de Vera, s/n, 46022 - Valencia (SPAIN)
†División de Servicios Avanzados en Movilidad. Telefónica I+D, Madrid (SPAIN)
*Email: jorsasma@iTEAM.upv.es

ABSTRACT

This paper deals with the detection of shot changes in order to improve H.264 compression efficiency. This improvement is achieved by inserting intra frames when cuts occur and coding the rest of the frames using inter-frame prediction. In previous works, the proposed algorithm has demonstrated to be a fast and robust method for low delay and very low bitrate video coding, based on the comparison of the number of intra-coded macroblocks with two thresholds, one fixed and the other adaptive. In this paper, the optimization of the algorithm to be applied to high quality and low delay video coding is discussed and the results with Enhanced Definition Television format sequences (EDTV) are presented. This algorithm is also compared with another recent method based on the same measure, with results favoring our approach.

Index Terms— Shot detection, H.264 video coding, Enhanced Definition TV.

1. INTRODUCTION

Temporal segmentation of video data into shots is a pre-requisite in many applications: automatic video indexing and editing, old film restoration, video coding, etc. So, the development of shot detection algorithms has registered a great increase in the last decade. In the literature we can find several reviews of methods, such as [1,2,3].

The purpose of our method is to detect the scene changes in order to encode the first frame of each shot as an intra frame, since shot changes represent the best choice to insert key-frames in the video sequence. The aim is to encode the next frames of the new shot based on the first one via motion compensation and prediction (inter-frames or P-frames) in order to improve coding efficiency. Several measures have been used to detect the cuts, such as the Sum of Absolute Differences (SAD) [6] or the number of intra-coded macroblocks in a P-frame [4,5,7]. Our method, encoder-oriented, uses this last measure, and its main contribution consists of applying a set of two basic thresholds, one adaptive and the other fixed, over the number of intra-macroblocks, providing a high detection robustness. In previous works [4], the method was applied to a H.264 encoder [8] for low delay, low bitrate, low frame rate, and small format video coding (QCIF). In this paper, its optimization to adapt the algorithm to high quality and low delay conditions is presented. The results for EDTV sequences are discussed and the performance of our method is compared with another recent algorithm based on the same measure [5], obtaining better results.

The paper is organised as follows: in section 2, the algorithm is described and the selection of the different threshold values is discussed; the results and comparisons are presented in section 3 and, finally, section 4 provides some conclusions.

2. PROPOSED ALGORITHM

When a cut appears in a video sequence and the coder tries to encode that frame as a P-frame, coding efficiency and quality decrease. The main reason is the low correlation between the new frame and the last picture of the previous shot. The goal of the proposed algorithm consists of detecting the cut while the frame is being inter coded and recoding it as an I-frame, exploiting the correlation between frames belonging to the same shot. Note that we are not interested in the detection of gradual transitions, where correlation among frames is high and can be exploited by the encoder for prediction.

The proposed algorithm, described in [4], is summarized here for clarity purposes: a shot change is detected if the number of intra macroblocks in a P-frame is higher than an adaptive threshold, T_A, which matches the statistics of the sequence and takes into account its motion characteristics. It is obtained as $T_A = m_k + T_a$, where m_k is the weighted average of the number of I-MB in the P-frames previous to the shot and T_a is a percentage of the total number of macroblocks in a frame. The adaptive

[1] This work has been supported by the Polytechnic University of Valencia "Programas de apoyo a la investigación y desarrollo" PAID-06-06 and PAID-04-07.

threshold is further limited by a fixed limiter threshold, T_L, obtaining the final threshold $T=min(T_A, T_L)$.

The flow graph in Figure 1 summarizes the operation of the algorithm when the frame *k+1* is being coded: if a macroblock has been intra-coded, the counter IMB_{k+1} is increased and compared with the threshold *T*. If it is reached and the number of consecutive P-frames (*nf*) coded since the last I-frame is higher than an auxiliary parameter *N*, a cut is detected and the inter-coding is aborted, recoding the frame as an intra frame. In order to avoid the insertion of expensive intra frames too close in time, after a cut detection, a fixed span time is defined, with *N* being the number of frames to be coded during this span time: $N = fps \times span(ms)/1000$. During the span $nf < N$ and a security threshold T_S, with $T_S > T_L$, is active instead of the general threshold *T*. This security mechanism allows us to detect very clear and very fast cuts and to reduce the number of false detections near a real shot change.

If a cut has not been detected, the number of consecutive coded P-frames (*nf*) and the average number of intra macroblocks (m_k) are updated, as seen in the graph, where α is a memory parameter (see [4] for details). After that, the next frame is P-encoded, repeating the whole detection process.

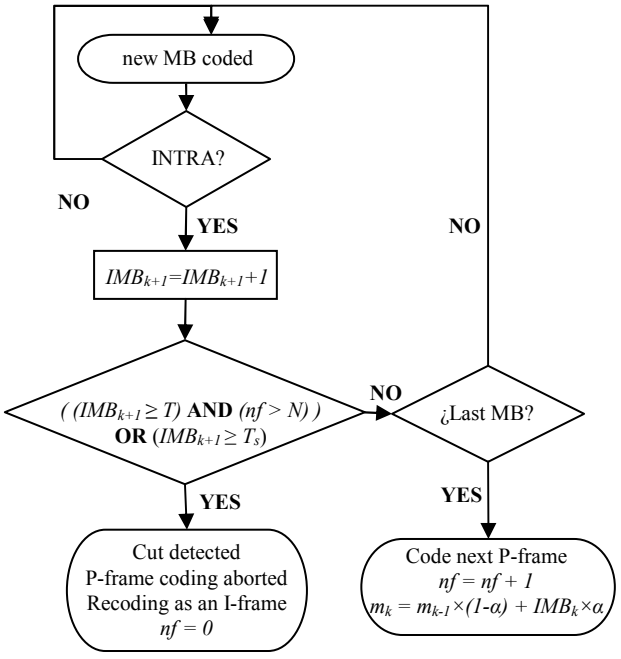


Figure 1: Flow graph of the cut detector algorithm when the k+1 P-frame is being coded.

Once the algorithm has been described, we are going to discuss the selection of the best values for the different parameters involved, now in high quality video coding. With the purpose of setting these parameters and checking the performance of the algorithm, two sets of video sequences have been used. A training set is used to fix the best values for the algorithm parameters and a test set is used to check the behaviour of the algorithm once the parameters have been fixed. The selected video sequences have been extracted from commercials and movies, and the real position of cuts has been manually extracted. To cover the widest possible range of shot changes, both sets have been selected depending on the motion content of the scenes and divided into four categories:

- Low Motion & Low Camera motion (**LM&LC**): dialog-like scenes with a steady camera: Pulp Fiction (frames 25300-30300 and 202000-207000), The Passion of the Christ (55000-60000) and Pirates of the Caribbean (418-5418): 385 cuts.
- Moderate Motion & Moderate Camera motion (**MM&MC**): smooth camera movement with more complex motion on scene: Final Destination II (10000-15000), Hero (10000-15000), Matrix (70400-75400 and 145000-150000) and Matrix Reloaded (frames 121200-126200): 461 cuts.
- High Motion & High Camera motion (**HM&HC**): action scenes with fast camera motion: King Arthur (10000-15000), Kill Bill (107007-112007), The Two Towers (15000-20000) and The return of the King (47500-52500 and 58000-63000): 477 cuts.
- All conditions (**ALL**): commercials, fast shot changes and gradual transitions (435 cuts).

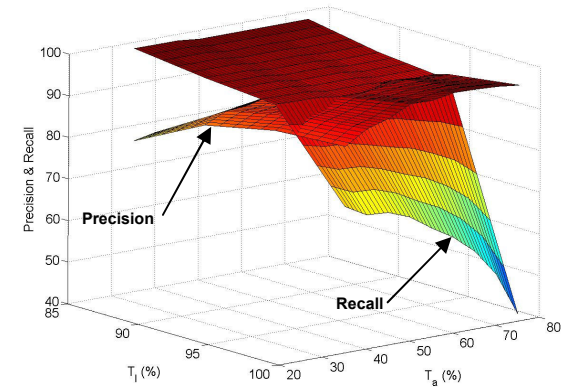
The shot detection algorithm has been applied to high quality, constant bitrate, H.264 EDTV video coding. For this purpose, three different configurations have been selected: 750 kbps for low bitrate, 1500 kbps for medium bitrate and 2000 kbps for high bitrate situations. With regard to the frame rate, 25 frames per second (*fps*) have been coded within all the bitrate configurations. Furthermore, a high quality configuration with lower frame rate has been selected, using 750 kbps and 12.5 frames per second. In order to preserve the low delay condition, an IPPPP GOP structure has been adopted.

To fix the values of its parameters and to analyse the behaviour of the algorithm, the number of missed detections (*MD*) and false alarms (*FA*) have been considered, compared to the total number of cut detections (*D*). These numbers are expressed as Precision (*P*) and Recall (*R*) [1,3].

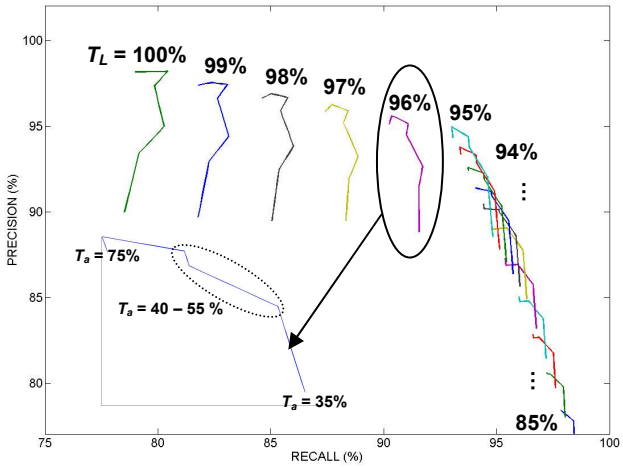
Different combinations of T_a and T_L have been tested to obtain their optimal values in terms of *P* and *R* (using fixed values for *N* and T_S). Figure 2 (a) shows the typical behaviour of *P* and *R* when T_a and T_L vary. As the thresholds increase, precision increases but recall decreases and vice versa, so the best possible values for the thresholds are those which obtain similar and high values for *P* and *R* at the same time. I.e., the higher intersection of both graphs shown in Figure 2 (a), obtained through an automatic searching process in the *P&R* mesh.

Figure 2 (b) shows again the relation between precision and recall for the different values of the thresholds. In this figure, the best value for the fixed threshold is reached near the upper-right corner, where higher values of *P* and *R* are obtained simultaneously. In this example, the best values are around $T_L = 95\%$.

In the detailed graph of the lower-left corner, the behaviour of the algorithm when the fixed threshold is 96% is shown. Note that the best values are obtained when the adaptive threshold ranges from 40% to 55%.



(a)



(b)

Figure 2: Selection of the thresholds: (a) precision and recall as a function of the thresholds, (b) relation between precision and recall with different values of the fixed threshold T_L. HM&HC example: 5000 frames at 25 fps and 1500 kbps.

FPS	Bitrate (kbps)	T_a (%)	T_L (%)	T_S (%)	Span (ms)		α (%)
					ALL	REST	
25	2000	55	96	98	350	500	40
	1500	50					
	750	45					
12.5	750	40					

Table 1: Parameters of the algorithm.

In Table 1 we present the final selection of the parameters of the algorithm, obtained statistically to produce the best possible global results in terms of P and R for the different configurations and video groups. The training video set used to fix these parameters consists of seven different video sequences with a total of 35000 frames and 818 shot changes. It must be noted that both T_L and T_S have been fixed to 96% and 98% respectively, based on the best average result for all the training video sequences, whereas the adaptive threshold, T_a, depends on the bitrate and the frame rate, varying from 40% to 55%. This main threshold increases with the bitrate and the frame rate, getting values slightly lower than 50% for lower bitrates and slightly higher values when the bitrate or the frame rate increase.

Once T_a and T_L have been fixed, the span value is adapted, and its optimal value has resulted to be 500 milliseconds, applying the security threshold during a constant gap of N=12 frames after each shot detection (at 25 frames per second). If the video sequence is known to have very fast shot changes, as in commercials or other complex video material, a lower value of the span time should be selected (a 350 ms span time is used to encode the video sequences of the ALL video category).

Finally, a memory parameter α = 40% has been selected based on previous simulations. As it can be seen, these results are consistent with the ones obtained in [4], where an adaptive threshold slightly lower than 50% is used for very low bitrate, low frame rate and small format.

3. RESULTS

In this section we present the results obtained when the final shot detection algorithm is applied to a battery of new test video sequences, consisting of 10 video sequences (50000 frames) and 876 shot changes.

		LM&LC		MM&MC	
FPS	bitrate	P (%)	R (%)	P (%)	R (%)
25	2000	98.90	100	99.43	98.92
	1500	99.43	100	98.89	98.93
	750	99.42	99.42	99.44	96.19
12.5	750	100	100	93.14	99.05
		HM&HC		ALL	
FPS	bitrate	P (%)	R (%)	P (%)	R (%)
25	2000	98.72	97.54	82.89	98.30
	1500	98.48	96.05	82.11	98.03
	750	98.43	93.47	81.34	97.81
12.5	750	96.16	96.25	82.60	96.86

Table 2: Precision and Recall for the different categories of video sequences in the test set.

Table 2 shows these results: the average value is over 94% for precision and over 95% for recall. In the best case the results are always over 98.9%, while in the worst case (HM&HC) the average value is higher than 96%. The special transitions such as fades, dissolves and flashes and the fast shot changes of the last category (ALL) produce a higher number of false detections, decreasing precision but maintaining high values for the recall parameter.

The typical local behaviour of our algorithm is shown in Figure 3, where five shot change detections are presented. The number of intra MB's is plotted with a solid line, while the general threshold (T) appears as a dashed line. The figure represents the coding of a MM&MC video sequence encoded at 750 kbps and 25 frames per second. The five detections (marked with a cross) correspond with shot changes (each one represented as a big dot): the shot change in frame number 4563 is detected with a threshold of about 50%, corresponding with a low motion scene; on the other hand, the cut in frame 4667 is detected with a 96% of intra MB's, showing the behaviour of the algorithm in high motion scenes.

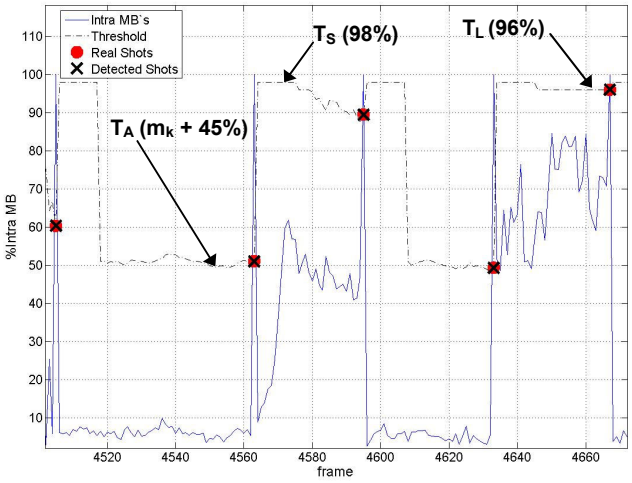


Figure 3: Number of intra MB's (solid) and threshold (dashed). Five shot change detections are shown, corresponding with shot changes.

	LM&LC	MM&MC	HM&HC	ALL
t_I / t_P	0.9	1.03	1.10	1.05
$(T_{ND}-T_D)/T_{ND}$	3.02%	2.36%	1.7%	2.2%

Table 3: Processing time results.

In this figure, the adaptive threshold, T_A, produces the detection in slow motion scenes, where the average number of intra MB's is low; the fixed threshold, T_L, is active when the adaptive threshold exceeds the 96% of the total number of MB's of a frame, and the security threshold, T_S, is active during the span time.

Table 3 shows the processing time results of the algorithm. The first row represents the relation between the average time used to detect a cut and recode it as an I-frame (t_I), and the average time needed to encode the same shot change as a P-frame (t_P). Due to the complexity of inter-coding a cut, a maximum coding time increase of only 10% in detecting and recoding a shot change frame is obtained. The second row shows the relative gain between the total time used to encode a complete sequence using the shot detection algorithm (T_D) and the total time needed to encode the same sequence without our algorithm (T_{ND}). Similar results to [4] have been achieved, obtaining a time gain of up to 3% in the case of LM&LC. These results prove that the algorithm is suitable to be applied to low delay video coding, even speeding slightly up the whole encoding process and obtaining a local gain in PSNR, showing the same behaviour as in [4].

Finally, Table 4 shows the precision and recall results when the algorithm described in [5] is used. This algorithm is also based on two thresholds: a fixed one and an adaptive threshold following the difference of intra MB's with respect to the previous frame. In comparison to this method, our algorithm reaches better detection rates, obtaining an average gain of 7% in precision and a 3% better recall. In certain situations (MM&MC), a precision gain of up to 16% and a 6% better recall are obtained.

		LM&LC		MM&MC	
FPS	bitrate	P (%)	R (%)	P (%)	R (%)
25	2000	98.28	100	83.05	92.31
	1500	97.58	100	83.18	92.97
	750	96.92	100	83.43	93.77
12.5	750	98.43	100	94.48	92.22
		HM&HC		ALL	
FPS	bitrate	P (%)	R (%)	P (%)	R (%)
25	2000	96.37	87.50	74.66	93.13
	1500	95.71	87.58	74.92	93.84
	750	96.62	94.07	75.80	95.23
12.5	750	89.83	92.98	77.28	92.27

Table 4: Precision and Recall for the different categories of video sequences using the algorithm from [5].

4. CONCLUSIONS

A fast and robust shot detection method for low delay, high quality H.264 video coding has been provided. Focusing on compression efficiency, the simple, two thresholds algorithm presented in [4] has been extended to EDTV H.264 coding with different configurations and similar high results. These final results, obtained after the analysis of nearly 1700 cuts, show the benefits of the algorithm: on the one hand, it achieves high detection rates, a low false alarm ratio in almost all the considered situations and a local PSNR gain around the detected shot change frames. On the other hand, the good processing time results make the algorithm suitable for low delay conditions. Finally, the algorithm has been compared with another recent method, based on the same measure, obtaining a better performance when our method is used.

10. REFERENCES

[1] P. Bouthemy, M. Gelgon and F. Ganansia, "A unified approach to shot change detection and camera motion characterization", *IEEE Trans. on Circuits and Systems For Video Technology*, Vol. 9, No.7, pp. 1030-1044,October, 1999.

[2] R. Brunelli, O. Mich and C.M. Modena, "A survey on the automatic indexing of video data", *Journal of Visual Comunication and Image Representation,* Vol. 10, pp. 78-112, 1999.

[3] C. Cotsaces, N. Nikolaidis and I. Pitas. "Video shot detection. A review", *IEEE Signal Processing Magazine,* Vol. 23, pp. 28-37, 2006.

[4] J. Sastre, P. Usach, A. Moya, V. Naranjo and J.M. López, "Shot detection method for low bit-rate H.264 video coding", *14th European Signal Processing Conference,* 2006.

[5] S. Spinsante, E. Gambi, F. Chiaraluce, "An improved error concealment strategy driven by scene motion properties for H.264/AVC decoders", *14th European Signal Processing Conference (EUSIPCO 2006),* 2006.

[6] A. Dimonu, O. Nemethova and M. Rupp, "Scene change detection for H.264 using dynamic threshold techniques", *5th EURASIP Conference*, 2005.

[7] G. Gennari, G.A. Mian and L. Celetto, "A H.264 robust Decoder with error concealment capabilities", *ST Journal of Research*, Vol. 2, No. 1, pp. 67-82, 2005.

[8] T. Wiegand and G.J. Sullivan, "The H.264/AVC Video Coding Standard", *IEEE Signal Processing Magazine*, Vol. 24, No. 2, March 2007.