Automatic shot boundary detection combining color, edge, and motion features of adjacent frames

A. Jacobs, A. Miene, G. T. Ioannidis, and O. Herzog

TZI - Center for Computing Technologies, University of Bremen,
Am Fallturm 1, D-28359 Bremen, Germany
{jarne|andrea|gtis|herzog}@tzi.de

Abstract

This paper describes the contribution of the Center for Computing Technologies (TZI), University of Bremen, Germany, to the shot detection task of the TREC 2004 video analysis track (TRECVID). The approach uses RGB histogram values which are calculated within a five frames width window and the edge change ratio between consecutive frames as well as frames at a distance of 10. Both methods are used to detect hard cut candidates. To confirm or reject the hard cut candidates a block-based motion analysis is used. Gradual transitions are also detected using RGB histogram values together with a finite state machine.

1 Introduction

The Center for Computing Technologies (TZI), University of Bremen, Germany, participated in the shot detection task of the video analysis track. In addition to 2003, where color histograms were also used, we calculate here not only differences between consecutive frames but also between pairs of frames with a distance of up to five frames. We added also a flashlight detection to avoid false alarms due to flashlights and we integrated the edge change ratio as an additional indicator for hard cuts. To avoid false alarms concerning hard cuts due to camera motion we integrated a block-based motion analysis to confirm or reject hard cuts detected by the color histogram or the edge change ratio method. For detecting gradual transitions we use a finite state machine.

2 Shot detection approach

This section introduces our 2004 shot detection approach. Section 2.1 describes the values calculated from the color histograms of six consecutive frames. In section 2.1.1 we define a condition, how false alarms caused by flashlight can be avoided. Section 2.1.2 describes the shot detection based on the RGB histogram values and the finite state machine. Hard cuts that start or end with a black frame are classified as fade in or fade out effects (sec. 2.1.3). Section 2.2 introduces the hard cut detection based on the edge change ratio (ECR). Section 2.3 shows how hard cut candidates are confirmed or rejected using motion information.

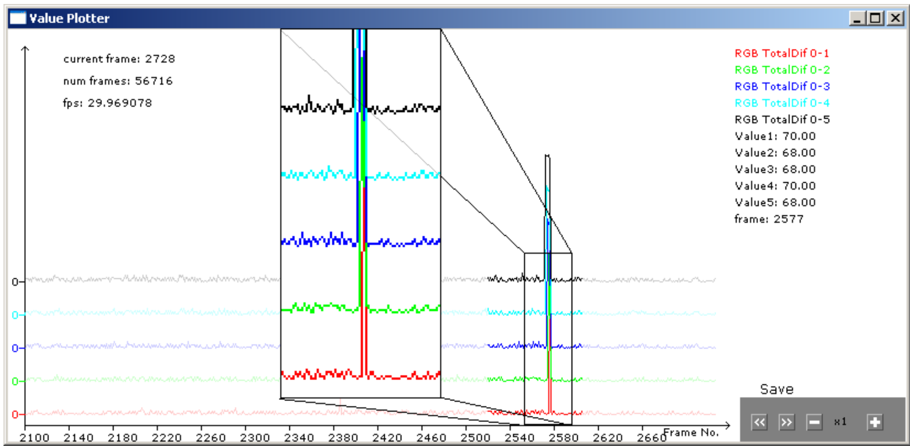

Figure 1: Plot of monitored values in case of a cut.

2.1 RGB histogramm

RGB histogram values are used to detect hard cut candidates and gradual cuts. First, we compute the average values for each color channel in each frame and compare these values in a window of five frames width, i.e. the actual frame is compared with frames that are in a distance of one up to five frames.

$$\Delta_{RGB}(n)_k = |R_n - R_{n+k}| + |G_n - G_{n+k}| + |B_n - B_{n+k}|. \tag{1}$$

where R_n, G_n, B_n is the average R, G and B value of frame n and $k = 1, \ldots, 5$ specifies the absolute RGB difference between the compared frames. Fig. 1 and 2 show plots of monitored values for the five frame distances in case of a hard cut and a gradual transition, respectively.

Then we compute the sum of the five difference values:

$$Dif\_Sum(n) = \sum_{k=1}^{5} \Delta_{RGB}(n)_k \tag{2}$$

In addition, we compute the sum of the squared differences of consecutive value pairs (with wrap around):

$$\begin{aligned} Sqr\_Dif\_Sum(n) \quad = \quad & (\Delta_{RGB}(n)_1 - \Delta_{RGB}(n)_2)^2 + \\ & (\Delta_{RGB}(n)_2 - \Delta_{RGB}(n)_3)^2 + \\ & (\Delta_{RGB}(n)_3 - \Delta_{RGB}(n)_4)^2 + \\ & (\Delta_{RGB}(n)_4 - \Delta_{RGB}(n)_5)^2 + \\ & (\Delta_{RGB}(n)_5 - \Delta_{RGB}(n)_1)^2 \end{aligned} \tag{3}$$

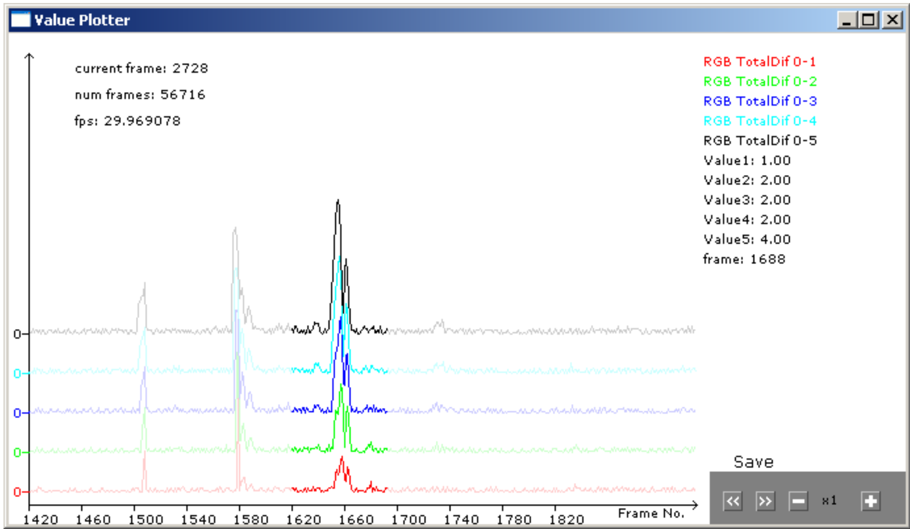

Figure 2: Plot of monitored values in case of a gradual transition.

2.1.1 Flashlight detection

To detect flashlights with a duration of maximum three frames we use the five absolute difference values $\Delta_{RGB}(n)_1, \ldots, \Delta_{RGB}(n)_5$ and compare them to the thresholds th_flash_low and th_flash_high. The threshold values were varied as listed in table 1. If the following condition holds, a flashlight is detected:

$$\begin{aligned} &\Delta_{RGB}(n)_1 < th\_flash\_low \wedge \\ &\Delta_{RGB}(n)_5 < th\_flash\_low \wedge \\ &(\Delta_{RGB}(n)_2 > th\_flash\_high \vee \\ &\Delta_{RGB}(n)_3 > th\_flash\_high \vee \\ &\Delta_{RGB}(n)_4 > th\_flash\_high) \end{aligned} \quad (4)$$

The frames that are illuminated due to the flashlight are skipped.

2.1.2 Shot boundary detection

A hard cut candidate is detected if the following condition holds:

$$\Delta_{RGB}(n)_1 \geq th_{RGB} \wedge \Delta_{RGB}(n)_5 \geq th_{RGB} \quad (5)$$

Decreasing the threshold th_{RGB} results in more hard cut candidates. Therefore, decreasing th_{RGB} increases the recall but decreases the precision of the detection of hard cuts. The hard cut candidates are then tested via block based motion analysis and are either confirmed, so that a hard cut is detected or rejected.

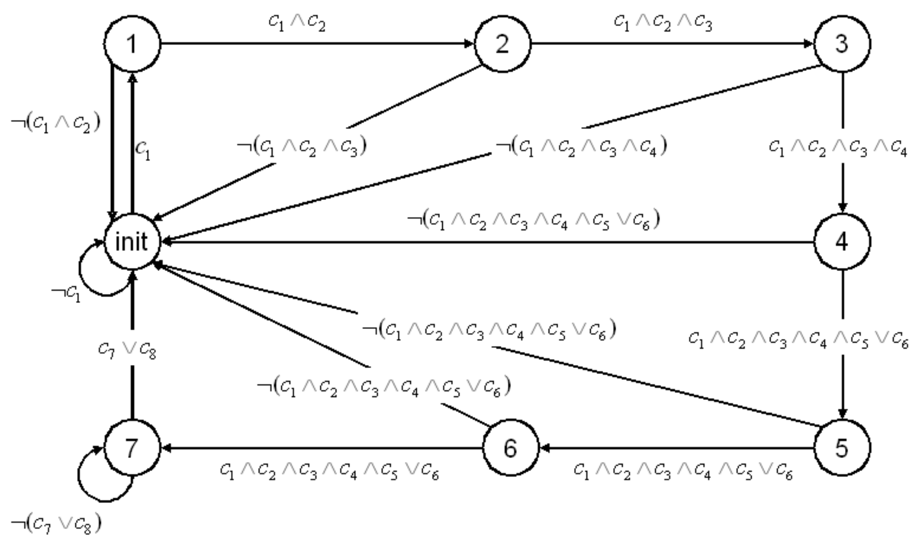

Figure 3: Finite state machine

For detecting gradual transitions and further hard cut candidates we use a finite state machine (FSM) with eight states comparable to that proposed in [Amir et al., 2003]. The FSM is illustrated in fig. 3. The five RGB mean difference thresholds ($th_{0-1}, \ldots, th_{0-5}$) control the conditions for changing the states. In case of a hard cut the absolute difference values $\Delta_{RGB}(n)_1, \ldots, \Delta_{RGB}(n)_5$ are significantly higher. For a gradual transition they change more slightly. Decreasing the five thresholds increases the recall but decreases the precision of the shot boundary detection.

The conditions for changing the states depend on combinations of the following comparisons of values against thresholds:

- $c_1 : \Delta_{RGB}(n)_1 \geq th_{0-1}$
- $c_2 : \Delta_{RGB}(n)_2 \geq th_{0-2}$
- $c_3 : \Delta_{RGB}(n)_3 \geq th_{0-3}$
- $c_4 : \Delta_{RGB}(n)_4 \geq th_{0-4}$
- $c_5 : \Delta_{RGB}(n)_5 \geq th_{0-5}$
- $c_6 : Sqr\_Dif\_Sum(n) > th_{sqrDifHigh}$
- $c_7 : Sqr\_Dif\_Sum(n) \leq th_{sqrDifLow} \wedge Dif\_Sum(n) \leq 300$
- $c_8 : Dif\_Sum(n) \leq th_{DifSum}$ with $th_{DifSum} \leq 300$

When changing from state 5 or 6 to the initial state a hard cut candidate is detected, which is then tested using the block-based motion analysis method (section 2.3).

When changing from state 6 to state 7 we save the previous frame as the possible beginning of a gradual transition. During the gradual transition the FSM stays in state 7. There are two possibilities to change from state 7 to the initial state and to detect the end of a gradual transition: First, when the absolute difference values do not have a significant variance, it is assumed that the frames belong to the same shot and the gradual transition is over. This is tested by the condition $c_7 = Sqr\_Dif\_Sum(n) \leq th_{sqrDifLow} \wedge Dif\_Sum(n) \leq 300$. So increasing the parameter $th_{sqrDifLow}$ will decrease the average length of the detected gradual transitions. This parameter only affects the precision. Second, when the condition $c_8 = Dif\_Sum(n) \leq th_{DifSum}$ with $th_{DifSum} \leq 300$ is fulfilled, then one expects having frames that belong to the same shot. Increasing th_{DifSum} leads to shorter gradual shots and therefore only affects precision. Otherwise no transition is detected. In both cases the detection process starts again from the beginning (initial state). Hard cuts which were detected during the gradual transition are rejected.

2.1.3 Fade in and fade out effects

Having detected a hard cut we test for black frames to find fade in and fade out effects. If the previous frame is black we declare the last detected shot as a gradual one and adapt the post frame (fade out). If the post frame is black we mark this frame and declare the next detected shot as a gradual one with the marked black frame as the pre frame (fade in).

2.2 Edge change ratio

This technique relies on the fact that the edges of the objects within the frames would definitely change across a boundary. In other words, temporal visual discontinuity always comes with structural discontinuity [Lienhart, 2001]. Exploiting the above fact, the percentage of edges that enter and exit between the two frames is computed.

The Edge Change Ratio $ECR(n,k)$, between the frames $n-k$ and n is calculated as shown below[Lienhart, 2001],

$$ECR(n,k) = max(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-k}^{out}}{\sigma_{n-k}}) \tag{6}$$

where, σ_n is the number of edge pixels in the frame n, and X_n^{in} and X_{n-k}^{out} are the entering and exiting edge pixels in frames n and $n-k$ respectively.

The edge pixels in one image that have edge pixels very close by (around 6 pixels) in the second image are not regarded as entering or exiting edge pixels to compensate for motion [Lienhart, 2001]. The technique was implemented based on [Lienhart, 2001]. The algorithm is as follows:

1. Convert frames $n - k$ and n to grayscale.
2. Perform Canny edge detection on these frames [Canny, 1986].
3. Count the number of edge pixels: σ_n and σ_{n-k}
4. Dilate the edges and invert the images.
5. Perform AND operation between the edge-image (frame $n - k$) from step 2 and the output image (frame n) from step 4. Also between edge-image (frame n) from step 2 and the output image (frame $n - k$) from step 4.
6. Count the number of entering and exiting edge pixels in the images from step 5 to obtain X_n^{in} and X_{n-k}^{out}
7. Calculate the $ECR(n, k) = max(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-k}^{out}}{\sigma_{n-k}})$

With $k = 1$ we obtain the ECR for consecutive frames $n, n - 1$. During the shot boundary detection we also use the ECR with $k = 10$, i.e. for frames at a distance of 10.

To detect hard cuts, two values are calculated: First, the near far ratio that describes the ratio between the ECR between the current and the next frame (near ECR) and the ECR between the current and the 10th frame (far ECR):

$$r_{nearfar}(n) = \frac{ECR(n, 10)}{ECR(n, 1)} \tag{7}$$

Second, we calculate the far last-far ratio which describes the ratio between the current far ECR and the previous far ECR values:

$$r_{farlastfar}(n) = \frac{ECR(n, 10)}{ECR(n - 1, 10)} \tag{8}$$

If $r_{farlastfar}(n) > 2.0 \vee (r_{farlastfar}(n) > th_{ECR_{farlastfar}} \wedge r_{nearfar}(n) > th_{ECR_{nearfar}})$ a hard cut candidate is detected, which is tested via block-based motion analysis and either is confirmed or rejected. Decreasing these thresholds leads to more hard cut candidates and therefore increases recall but decreases precision.

The test via block-based motion analysis is omitted if $r_{nearfar}(n) > 5.0$, because this is definitely due to a hard cut.

Fig. 4 shows a plot of monitored ECR values in case of a hard cut.

2.3 Block-based motion analysis

Hard cut candidates are tested via a block-based motion analysis for being confirmed or rejected. We compute a motion vector field and sort the vectors into 12 bins, each representing a direction interval in 2D covering 30°. If the vectors are spread over all bin (i.e. there are no real peaks) a hard cut is confirmed.

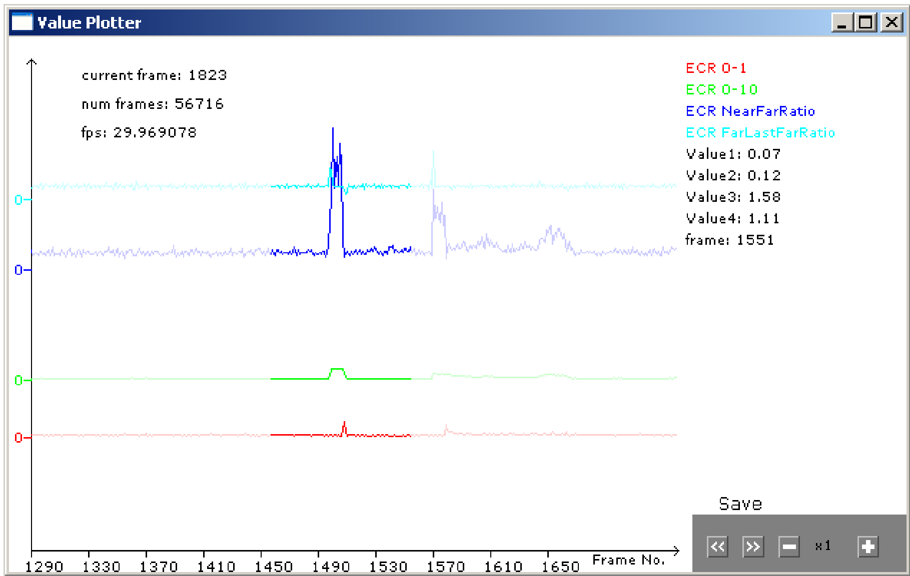

Figure 4: Plot of monitored values in case of a cut.

3 Results

Table 1 lists the parameter settings for each of the 8 runs submitted.

Table 2 lists the results measured by precision and recall for each run.

Acknowledgement

We would like to thank Alexander Köhn and Srikanth Nagandla for their contribution during the implementation of this approach.

Part of this work was supported by the DELOS Network of Excellence on Digital Libraries, funded by the EU's Sixth Framework Programme.

References

[Amir et al., 2003] Amir, A., Berg, M., Chang, S.-F., Hsu, W., Iyengar, G., Lin, C.-Y., Naphade, M., Natsev, A. P., Neti, C., Nock, H., Smith, J. R., Tseng, B., Wu, Y., and Zhang, D. (2003). IBM research TREC-2003 video retrieval system. In *Proceedings of TREC Video Retrieval Evaluation (TRECVID 2003). Gaithersburg, MD, NIST.*

[Canny, 1986] Canny, F. J. (1986). A computational approach to edge detection. *IEEE Trans PAMI*, 8(6):679–698.

[Lienhart, 2001] Lienhart, R. (2001). Reliable Transition Detection In Videos: A Survey and Practitioners Guide. *International Journal of Image and Graphics (IJIG)*, 1(3):469–486.

Parameter	Run 1	Run 2	Run 3	Run 4
th_flash_low	12	10	10	10
th_flash_high	20	20	30	30
RGB mean difference				
th_{0-1}	9	9	11	8
th_{0-2}	10	11	11	10
th_{0-3}	12	12	12	14
th_{0-4}	14	14	14	16
th_{0-5}	14	18	14	18
$th_{sqrDifLow}$	20	20	30	40
$th_{sqrDifHigh}$	3000	1000	2000	1000
th_{DifSum}	20	20	30	40
th_{RGB}	35	30	45	35
$th_{ECR_{nearfar}}$	1.80	1.70	1.70	1.70
$th_{ECR_{farlastfar}}$	1.50	1.40	1.45	1.50

Parameter	Run 5	Run 6	Run 7	Run 8
th_flash_low	10	10	10	10
th_flash_high	20	30	30	18
RGB mean difference				
th_{0-1}	8	12	8	7
th_{0-2}	10	12	12	10
th_{0-3}	14	12	20	15
th_{0-4}	16	12	30	20
th_{0-5}	18	12	40	30
$th_{sqrDifLow}$	20	20	150	60
$th_{sqrDifHigh}$	3000	1000	8000	10000
th_{DifSum}	20	20	40	30
th_{RGB}	30	40	50	60
$th_{ECR_{nearfar}}$	1.90	1.80	1.85	1.70
$th_{ECR_{farlastfar}}$	1.50	1.50	1.60	1.45

Table 1: Parameter settings.

	All		Cuts		Gradual			
Run	Recall	Prec.	Recall	Prec.	Recall	Prec.	Frame-R.	Frame-P.
1	0.719	0.634	0.896	0.636	0.347	0.621	0.506	0.619
2	0.736	0.615	0.895	0.620	0.401	0.595	0.531	0.637
3	0.664	0.642	0.836	0.650	0.302	0.601	0.491	0.729
4	0.742	0.610	0.906	0.614	0.394	0.592	0.501	0.799
5	0.725	0.626	0.895	0.628	0.367	0.619	0.512	0.630
6	0.732	0.647	0.893	0.654	0.394	0.617	0.545	0.637
7	0.655	0.639	0.853	0.632	0.236	0.698	0.439	0.820
8	0.704	0.632	0.889	0.625	0.313	0.674	0.455	0.858

Table 2: Evaluation results.