Linear Transition Detection as a Unified Shot Detection Approach

Costantino Grana, *Member, IEEE,* and Rita Cucchiara, *Member, IEEE*

***Abstract*— In this paper, we propose an automatic system for video shot segmentation, called Linear Transition Detector (LTD), unique for both cuts and linear transitions detection. Comparison with publicly available shot detection systems is reported on different sports (Formula 1, basket, soccer and cycling) and TRECVID 2005 results are also reported.**

I. Introduction

THE new generation of Video Digital Library Management Systems require tools for video segmentation into significant units and for structure annotation. Similarly to the structural description of the documents in textual digital libraries (DLs), the annotation of the structural units of video documents allows advanced processes of indexing, querying, retrieval, summarization, browsing and other types of accesses, as requested by the users. Typically, for edited videos, the basic significant structural units are the shots, which are sets of consecutive frames taken from a single camera in a single continuous operation.

Multimedia tools need general procedures which should possibly be data content independent, and also less sensitive to the amount of motion and color variability within the shot. Moreover, in edited videos many types of shot transitions (cuts, fades, wipes, etc.) can be found, while most of the detection techniques are devised for a single type only. Therefore, desirable methods should be flexible, with few parameters and unified approaches for different types of transitions.

This work proposes a new two step iterative algorithm, which relies on a linear transition model, also able to identify transition center and length. Its performance is compared with state of the art approaches in terms of recall/precision and experiments are reported, carried out in three different contexts: the MPEG-7 Content Set Sports Videos [1], part of a Formula 1 DL and on TRECVID 2005 videos.

After an overview of the state of the art in the next section, in Section III we describe the shot and transition segmentation algorithm, while in Section IV we show the comparison with other approaches presented in academic research.

II. Related Work

In recent years many techniques have been proposed for the detection of abrupt transitions (hereinafter *cut*), and they have proved to give highly satisfactory results. The most common ones exploit differences of some metrics between adjacent frames [2]. Some of them have addressed the problem of shot detection in the compressed domain; an example is the work of Pei and Chou [3]. The only information extracted from the videos in the compressed domain approaches are those directly available from the MPEG streams, that is DCT coefficients, motion vectors and directions of prediction for each block. The absence of the decoding process allows a much faster computation, but has the drawback of having a lower reliability, especially in the presence of high motion. The main problem in gradual transition detection is that the classical comparison based on adjacent frames is not suitable because inter-frame changes are too small. Therefore, alternative approaches analyze large windows, i.e. they compute the difference between a frame and the k-th following one, instead of the next. The analysis of a large window is necessary to include the whole transition, but at the same time it is not trivial since the variation between two different shots can be confused with the motion variation within the shot. The algorithm proposed by Yeo and Liu [4] tries to find a "plateau" in the difference values extracted with a single fixed frame-step. A more refined approach is proposed by Heng and Ngan [5], where the authors deal with long transitions. Each frame is compared against a reference frame chosen from the sequence, and a one-dimensional change indicator is computed. This indicator behaves as a "ramp" during a transition and is constant in the same shot. The authors estimate the slope of the ramp and the standard deviation at the border to find transitions boundaries. A very similar method is proposed by Huang and Liao [6], where each frame is compared against a fixed one termed a "seed". When the window is long enough, difference values are uncorrelated, and a non decreasing ramp of values could indicate a transition.

In [7], Bescos discloses a very accurate comparative study of most of the metrics used in shot detection approaches, both in the compressed and uncompressed domain and proposes an algorithm to detect both abrupt and gradual transitions which has a good degree of generality, handling all types of transitions. The transition is detected describing its temporal evolution pattern with a set of rules, e.g. the maximum value of the window should be located in the center, the magnitude of the central disparity value should be reached gradually, etc. Here, the frame differences are computed using multiple frame-steps, and thus the final decision space is given by multiple feature set, one for each frame step. Recently, the same author proposed a unified framework [8] for both cuts and transitions, that showed very good results. The drawback of this method is that 20 parameters are needed (10 for cuts

C. Grana and R. Cucchiara are with the Department of Information Engineering of the University of Modena and Reggio Emilia, Italy.

This work was supported by the DELOS Network of Excellence on Digital Libraries, as part of the IST Program of the European Commission (Contract G038-507618).

and 10 for transitions), thus making the training process more complex.

The authors of [9] have developed a neural network classifier to detect transitions. The classifier is trained with a dissolve synthesizer which creates artificial dissolves. The algorithm works on contrast-based features, as well as color-based features, and has given good results compared to standard approaches based on edge change ratio. A training process is also required in [10], where a probabilistic based algorithm is proposed for detecting both abrupt and gradual transitions. After constructing a-priori likelihood functions by means of training experiments, they take into account all relevant knowledge to shot boundary detection, like shot-length distribution and visual discontinuity patterns at shot boundaries.

While the aforementioned works are based on frame differences and address gradual transitions of any type, another widely explored technique is the one proposed in [11], which only addresses linear transitions. The property exploited therein is that the mean and variance of pixel intensity during the transition have a linear and quadratic behavior, respectively. Therefore the criterion used to determine the presence of a transition is that the ratio of the second derivative of the variance curve to the first derivative of the mean curve should be a constant.

Our approach is strictly focused on gradual transitions with a linear behavior, including abrupt transitions. A precise model is exploited which allows to achieve more discriminative power than with general techniques. We developed an iterative algorithm which, given a frame of possible transition, alternatively tries to find the best center position for the transition and best length, by minimizing an error function, which measures the fitness of data to the linear model.

III. Linear Transition Detector

Before describing our algorithm of shot segmentation in detail (indicated throughout the paper as Linear Transition Detector, or LTD), it will be useful to define the ideal model of linear transition and to underline its important properties. These properties will be exploited by the algorithm to cope with non idealities and to measure the confidence of the detection.

A. The Transition Model

Let us consider two consecutive shots in a video sequence, the first one ending at frame e, and the second one starting at frame s, with $e < s$; $s = e+1$ implies an abrupt cut, otherwise a gradual transition is present.

To design a shot segmentation algorithm, two assumptions are necessary: the first one is that a feature $F(t)$ is computable at time t, with the characteristic of being both discriminating and almost constant within the shot; ideally

$$\begin{aligned} &F(t) = F(e), \forall t \leq e \\ &F(t) = F(s), \forall t \geq s \\ &F(e) \neq F(s). \end{aligned} \tag{1}$$

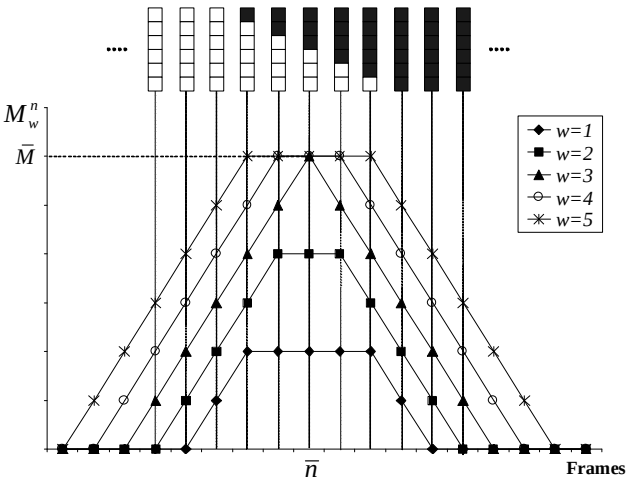


Fig. 1. Values of M_w^n for an ideal linear transition with $L = 5$.

The second assumption is that a distance function exists in the feature space $\Phi : d : \Phi \times \Phi \rightarrow \mathbb{R}$, which shows a constant behavior during the transition:

$$d(F(t), F(t-1)) = c \quad e < t \leq s \tag{2}$$

Sometimes there is confusion on the definition of the length of a transition, because one may include in the count the first frame of the new shot after the transition (e.g. [8]), or the last one of the previous one. In our model, the length is the number of frames in which the transition is visible, that is $L = s - e - 1$. Note that this model also includes abrupt cuts in the definition of transition. A cut is a transition with length $L = 0$. The transition center is defined as $\overline{n} = (e + s)/2$ and may correspond to a non-integer value, that is an inter-frame position. This is always an inter-frame position in case of cuts.

Differently from other formulations, instead of computing the difference between the frames $F(i)$ and $F(i + w)$, w being the *frame-step*, we calculate a metric M_w^n centered on frame or half-frame n, with $2n \in \mathbb{N}$, and with frame-step $2w \in \mathbb{N}$. It is defined as:

$$M_w^n = \begin{cases} d[F(n-w), F(n+w)] & n + w \in \mathbb{N} \\ \frac{1}{2}\left[M_w^{n-\frac{1}{2}} + M_w^{n+\frac{1}{2}}\right] & otherwise \end{cases} \tag{3}$$

The second term of the expression is a linear interpolation adopted for inter-frame positions. This is necessary because the feature F is relative to a single frame and cannot be directly computed at half-frames. The reason for expressing the metric as $d[F(n-w), F(n+w)]$ instead of $d[F(n), F(n+2w)]$ will be explained in section III-C.2.

In Fig. 1 an example of ideal linear transition with $L = 5$ is depicted, from a shot with completely white frames to one with completely black ones. If the transition is perfectly linear according with the hypothesis of Eq. 1 and Eq. 2, the shape of function M_w^n is an isosceles trapezoid centered in $\overline{n}$, for each possible window w. It degenerates into a triangle when $2w = L + 1$, since the minor base of the trapezoid becomes a point.

It is possible to verify that in this ideal case, given

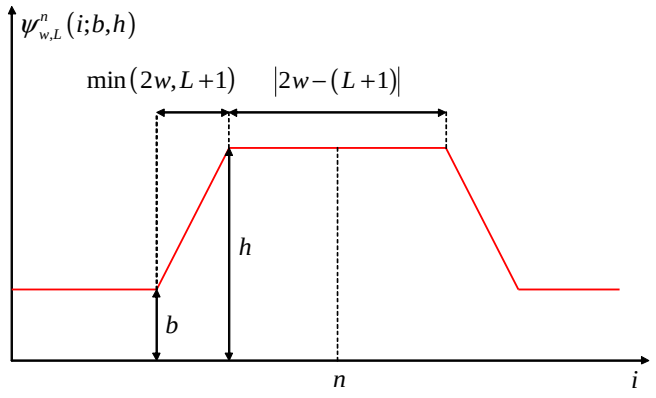


Fig. 2. Trapezoidal shaped function $\psi^n_{w,L}(i;b,h)$

the model of Eq. 1 and Eq. 2 and the metric of Eq. 3, both the up and down slopes are triangles with a base of $\min(2w, L+1)$ frames, and that the plateau of absolute maximum is $|2w-(L+1)|$ long (see Fig. 2). It's also straightforward to verify that:

$$\begin{array}{ll} M^{\overline{n}}_w < \overline{M}, & \text{if } 2w < L+1 \\ M^{\overline{n}}_w = \overline{M}, & \text{if } 2w \geq L+1 \end{array} \tag{4}$$

where $\overline{M} = \max_{w,n} M^n_w$ (see Fig. 1). We define as $\psi^n_{w,L}(i,b)$ the generic trapezoidal function, centered in n, whose value is M^n_w at the center (the absolute height of the minor base) and b is the value outside the trapezoid. The function is plotted in Fig. 2. We define $\psi^n_{w,L}(i) = \psi^n_{w,L}(i,0)$, the function which corresponds to the ideal transition case.

In an actual case, camera operations and objects motions, color and luminance variation cause the feature F to be non constant throughout the shot, thus making Eq. 1 and Eq. 2 not exactly satisfied. The consequence is that the shapes of both the slopes and the plateau are usually disturbed as affected by noise. The same effect occurs in the case of non linear special effects or wipes where Eq. 2 is not satisfied by definition. Nevertheless, for short transitions, the graduality of the variation can be considered linear in a first approximation. The selection of a linear feature $F(t)$ easy to compute is not straightforward. The choice of a feature which is not exactly linear, but is approximately linear in small windows, may introduce further imprecisions which can be accounted as noise, as the aforementioned factors. Therefore in this work, we assume to have a linear model, but provide an approach able to cope with all these non linear factors.

B. Choice of Features

The F feature and the M metric must be selected. In fact, a combination of metrics is often needed to achieve good results in shot detection: as in [2] we use a pixel-based metric and a histogram-based one. While the former is more sensitive to small changes in the image appearance and motion, the latter has higher stability but misses local information.

The sum of squared differences was employed as the pixel-based distance:

$$d_\delta(I_a, I_b) = \sum_{(x,y)\in I_{a,b}} [I_a(x,y) - I_b(x,y)]^2 \tag{5}$$

I_a and I_b being two images or in our case frames. The χ^2 test is used as the histogram based distance, which has been proved to perform better than other measures in many cases [6]:

$$d_{\chi^2}(I_a, I_b) = \sum_{c\in\{R,G,B\}} \sum_{j=1}^{N} \frac{(H_{c_a}(j) - H_{c_b}(j))^2}{\max(H_{c_a}(j), H_{c_b}(j))} \tag{6}$$

where $H_{c_a}(j)$ ($H_{c_b}(j)$) is the j-th bin value of the histogram of color plan c of the image I_a (I_b), and N is the number of bins. Introducing histograms in the feature space has of course the effect of adding a non-linear behavior to the feature space, but in most actual cases, the distance can be considered acceptably linear. It is very simple to construct toy examples of highly non-linear behavior, but in our tests on actual data this was kept within reasonable limits.

The two distances are combined in a single *difference measure* by means of two coefficients:

$$d(I_a, I_b) = c_{\chi^2} d_{\chi^2}(I_a, I_b) + c_\delta d_\delta(I_a, I_b) \tag{7}$$

The two coefficients are estimated from a training set of clips, using linear discriminant analysis, aiming at providing discrimination between frames belonging to the same shots or to different ones. In Eq. 3 the feature $F(\cdot)$ is computed at frames I_{n-w} and I_{n+w}. In particular we recorded all the distances which had been computed during the analysis of a whole video, then annotated each pair with same-shot or different-shots label, finally applying LDA. So the distances were computed at different frame steps, not just at consecutive distances.

C. Two Step Algorithm

In the ideal case a correlation between the data and the function $\psi^n_{w,L}$ could be a sufficient indicator of transition presence, but, due to lack of ideality in most of the shot transitions, we employ an algorithm constructed of two steps: the first one searches for the transition center position n, assuming a fixed frame step $2w$, and the second one searches for the transition length L by trying different values of w, but keeping fixed the transition center found in the previous step. While in the ideal case just the first step would be sufficient, in actual cases an error in locating the center position would also lead to a wrong estimate of the length. For this reason, a second step is introduced to provide a different view of the function behavior, to give a possible confirmation on the first step outcome and a new estimate for the window size. Iteratively repeating the two steps allows to progressively decrease the error. Then, a final verification step analyzes the decision space to validate as a transition the detected position.

In this section we explain the LTD algorithm in detail. We perform the following analysis considering a window of W frames, then we move the window forward $W/2$ frames and begin a new analysis. The W value limits the maximum

Algorithm 1 Linear Transition Detector

procedure LTD(W_0, W, T_P, T_E) ▷ W_0 is the first frame of the window, W is the window size
  $\overline{w} \leftarrow InitValue$
  repeat
    for $n \leftarrow W_0, W_0 + W$ **do**
      for all L that fit W given n **do**
        $\overline{n} \leftarrow \arg\max \Lambda^n_{\overline{w}L}$ ▷ Eq. 8
      end for
    end for
    for all $w < W$ **do**
      $\overline{w} \leftarrow \arg\min Z^{\overline{n}}_w$ ▷ Eq. 9
    end for
  until $2w = L + 1$
  compute $Peak^{\overline{n}}_{\overline{w}}$ ▷ Eq. 10
  compute $err^{\overline{n}}_{\overline{w}}$ ▷ Eq. 11
  if $Peak^{\overline{n}}_{\overline{w}} > T_P \wedge err^{\overline{n}}_{\overline{w}} > T_E$ **then**
    $Transition(\overline{n}, \overline{w}) \leftarrow TRUE$
  end if
end procedure

transition length and is directly related to the computational time of the process.

1) First Step: In the first step the values of M^n_w are calculated using the frame-step $\overline{w}$, which is found in the previous iteration of the algorithm, or is arbitrary chosen for the first iteration. The best trapezoid $\psi^n_{w,L}(i)$ is searched by moving the center n, and trying different values for L, while keeping $\overline{w}$ fixed. The trapezoid extends over $\delta = \min(2w, L+1) + |w - (L+1)/2|$ frames on the left and on the right of the center frame. For each couple of n and L the following matching measure is computed:

$$\Lambda^n_{\overline{w},L} \sum_{i=n-\delta}^{n+\delta} \min\left(M^i_{\overline{w}}, \psi^n_{\overline{w},L}(i)\right) - \sum_{i=n-\delta}^{n+\delta} \left|M^i_{\overline{w}} - \psi^n_{\overline{w},L}(i)\right| \tag{8}$$

The value of n is searched within the W frames window, and L must be varied such that $n+\delta$ and $n-\delta$ don't exceed the window.

In Eq. 8, two components are evident: the first one accounts for the area of the M^n_w function under the trapezoid, while the second component describes the similarity of our linear hypothesis with the data. It is very important to include both components, since we expect M^n_w to give a trapezoidal shape (the second term in Eq. 8), but we also require its *strength*, i.e. the amount of difference between the first and the second scene, to be significant. The first term in Eq. 8 in fact describes how large the trapezoid may be with respect to M^n_w and the second accounts for the adherence to the model. After finding the trapezoid which maximizes $\Lambda^n_{\overline{w},L}$, $\overline{n} = \arg\max_n \Lambda^n_{\overline{w},L}$ is the candidate transition center.

2) Second Step: Thanks to the definition of M^n_w as a distance function centered in n, as in Eq. 3, increasing the frame-step w makes the value of M^n_w rise to an absolute maximum when $w = (L+1)/2$ and then to be stable. It is easy to demonstrate that, in the ideal case, this growth is linear. Thus the growing function M^n_w against w graphically produces a linear slope followed by a horizontal line, when the value of M^n_w is stable. The second step of the algorithm uses this property to give an estimate of the transition length, by finding the smallest w which maximizes M^n_w. To provide a technique able to deal with noise, the tilt change of the chart is searched by minimizing the function:

$$Z^{\overline{n}}_w = \sum_{i=0}^{w} \left| M^{\overline{n}}_i - \frac{M^{\overline{n}}_w}{w} i \right| + \sum_{i=w+1}^{W} \left| M^{\overline{n}}_i - M^{\overline{n}}_w \right| \tag{9}$$

where W is the maximum size that a transition can assume. The w value that minimizes $Z^{\overline{n}}_w$ becomes our current frame step for the next iteration of the algorithm.

In typical conditions, the algorithm progressively narrows the trapezoids minor base leading to the expected triangular shape and finding the correct w and thus the transition length. Convergence is not guaranteed in non-ideal conditions and, for this reason, we add a convergence constraint: at each iteration the minor base of $\psi^n_{w,L}(i)$ is forced to become smaller down to zero. In Fig. 3 the M^n_w values are shown for 4 successive iterations of the algorithm in a real gradual transition case. At each iteration, we achieve a more precise estimate of the transition center and length, and thus a shape more similar to a triangle.

This second step tests with every possible value of w (according to W) in order to find the minimum value with maximum output. The advantage is that, since the second step is run just around a few selected points, the total number of computation is definitely less then the product of the ranges of L,n and w.

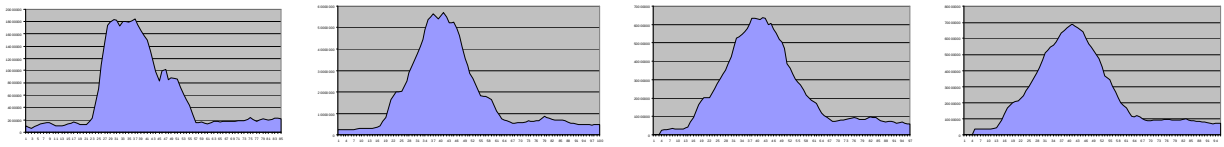
3) Decision Space: Given the transition length $L = 2w - 1$ and its center $\overline{n}$, as detected by the algorithm, the function $\psi^n_{w,L}(i)$ becomes triangular shaped. We must now verify the significance of the transition and how well the real data fits to the linear transition model. We introduce the following measure:

$$Peak^{\overline{n}}_{\overline{w}} = M^{\overline{n}}_{\overline{w}} - \min\left(M^{\overline{n}-2\overline{w}}_{\overline{w}}, M^{\overline{n}+2\overline{w}}_{\overline{w}}\right). \tag{10}$$

The $Peak^{\overline{n}}_{\overline{w}}$ value measures the height of the center value with respect to the lower of the two values of M in correspondence to the extremes of the triangle, and provides information on the transition significance. In fact, while in the model $M^{n\pm 2w}_w = 0$, in real cases this is not true, because of object and camera motion that causes the feature F to be not constant before and after the transition. To cope with this effect we have to get rid of the hypothesis of having an isosceles triangle and define the fitting error measure as:

$$\begin{aligned} err^{\overline{n}}_{\overline{w}} = \tfrac{1}{4\overline{w}} \sum_{i=1}^{2\overline{w}} & \left| M^{\overline{n}-i}_{\overline{w}} - \psi^{\overline{n}}_{\overline{w},L}(\overline{n}-i, M^{\overline{n}-2\overline{w}}_{\overline{w}}) \right| \\ & + \left| M^{\overline{n}+i}_{\overline{w}} - \psi^{\overline{n}}_{\overline{w},L}(\overline{n}+i, M^{\overline{n}+2\overline{w}}_{\overline{w}}) \right| \end{aligned} \tag{11}$$

The error sum is divided by the triangle's base $4\overline{w}$ to obtain a measure which is independent from the transition length. A minimum threshold on the $Peak^{\overline{n}}_{\overline{w}}$ value, T_P and a maximum threshold on $err^{\overline{n}}_{\overline{w}}$, T_E, are employed to discriminate real shot changes from false ones. The final decision space is then based on two parameters only which are the same for cuts



(a) (b) (c) (d)

Fig. 3. Four successive iterations of the algorithm in a real gradual transition: at each iteration, the shape of M_w^n values becomes more similar to a triangle

and transitions.

In conclusion, the algorithm evaluates at every window of size W each value $\overline{n}$ to verify if it is a possible point of transition whose length is given by $2w-1$. Since the algorithm is run on the same frames more than once (the windows are overlapped each other) a trivial check skips consecutive transitions detected twice. In this way the end of the current shot and the beginning of the next one are detected, so that the video may be annotated as a sequence of shots and transitions. Transition can have length $L = 0$ when $w = 1$, that is in case of cuts.

TABLE I
VIDEO SET

	Frames	Abrupt Cuts	Gradual Transitions	
			Dissolves	Effects
F1 Italy	124940	571	221	29
F1 Austria	138452	689	83	37
F1 Europe	153860	625	197	45
Soccer	22514	53	17	8
Basket	23361	75	26	12
Cycling	15407	2	43	0

IV. EXPERIMENTAL RESULTS

For our tests of the performance of shot segmentation against a ground truth, we used 3 full-length Formula 1 race videos and three clips from other sports (basket, soccer and cycling), taken from the MPEG-7 Content Set. All these videos showed a mixture of abrupt cuts and gradual transitions, namely dissolves and more complex editing effects. The number of frames, cuts and transitions for each video is shown in Table I. Table II shows the results of our algorithm in terms of precision and recall (P and R respectively). The results of precision and recall are affected by the choice of the couple of parameters T_E and T_P. In the tests, the thresholds T_E and T_P have been tuned using the video "Formula 1 Italy" as training set. We selected the thresholds, through exhaustive search, to maximize the sum of precision and recall.

In Fig. 4 different types of correctly detected transitions are shown. Row 1 shows an example of abrupt cut, row 2 an example of dissolve between almost static scenes. Row 3,4 and 5 show examples of special edit effects. All these effects are detected by our algorithm because they have a linear component included in their temporal evolution (see figure caption for more details). The fact that the transition is not wholly linear affects the frame recall rate, which is reduced to the linear component only. With "frame recall rate" we follow the terminology introduced by TRECVID, that is the number of transition frames really detected. In the examples, even if detected, the transitions are identified only in their linear part.

All videos have been manually segmented by locating cuts and gradual transitions, together with their length. The results are shown for cuts only, gradual transitions only, and overall. The results for cuts are obtained by discarding all the false negatives and true positives due to gradual transitions, and vice versa for the other case. Since the number of false positives is unchanged, the precision is always lower in the partial results than in the overall. The last column shows the mean of precision and recall. We can note that the results are similar and fairly good in all videos but Cycling. The sequence in question is very illuminated with many saturated zones. The problem is that we didn't train on such a sequence and thus when the graphic comment pops up (with dark background) it often causes false detections. A practical solution could be to reduce the feature computation area to avoid including this graphical comments, but this is not possible for a general purpose systems which doesn't want to be excessively tuned to a particular application.

To compare the results of our algorithm with publicly available systems we employed VCM [12], [13] by TZI (Technologie-Zentrum Informatik) and VideoAnnex [14] by IBM, which are freely downloadable shot detection software which provide MPEG-7 or XML formatted output.

In Table III we show the comparison between our algorithm and the others for three Formula 1 videos. Results are shown for separate cases of abrupt cuts and gradual transitions. The training process has been done on the first video. Our algorithm reaches the best overall results, in terms of average precision and recall, in all three videos.

Some remarks must be made about the algorithms used in this comparison: VCM and VideoAnnex don't need any training and there isn't any parameter that needs to be set manually. This is a great advantage from the user's point of view, but it may lead to worse results in the comparison, because the parameters cannot be tailored to the specific kind of videos. These two algorithms don't provide the length of the transitions in their output, but they only give the shot change frame, so we consider the shot change frame correct only if it is in correspondence of a transition in the ground truth (that is, in case of gradual transitions, included in the transition's frames).

By looking at Table III some observations can be done: the VideoAnnex algorithm has a recall for gradual transitions much higher than the other algorithms, but with very low precision (that means an oversegmentation of the video). Also its performance with abrupt cut is very poor, thus leading to

Fig. 4. Examples of correctly detected transitions. From top to bottom: cut, dissolve, 3 special edit effects, which are correctly detected because they contain a linear part. For instance, in effect on row 3 there is a dissolve, which is clearly visible at the end. A dissolve is also present at the beginning of effect on the last row. The effect on row 4 shows a linear transform in its last frames.

TABLE II

RESULTS IN TERMS OF PRECISION (P) AND RECALL (R) OF OUR ALGORITHM. VALUES ARE EXPRESSED AS PERCENTAGES

	Abrupt Cuts		Gradual Transitions		Overall	
	P	R	P	R	P	R
Basket	83	99	64	71	87	89
Soccer	96	83	88	56	97	74
Cycling	8	50	77	84	77	82
F1 Italy *	84	95	67	84	88	91
F1 Austria	86	96	50	87	88	95
F1 Europe	93	95	81	84	94	92

* *F1 Italy* has been used as training set

mediocre overall results. On the contrary, the VCM algorithm is good on detecting cuts, but has very poor results on gradual transitions.

The algorithm has also been tested at 2005 TREC Video Retrieval Evaluation (TRECVID2005), sponsored by the National Institute of Standards and Technology. The first task was to detect cuts and gradual transitions (described with their length) on a set of 11 videos. The videos were extracted from regular TV programming in Arabic, Chinese and English language sources. Training data was made available before the test data, so we selected the two thresholds based on one video for each source (LBC, CCTV4, NTDTV, CNN, NBC, MSNBC). In the end we observed that, differently from the sports videos, the Error threshold didn't add any benefit to the results, probably because of the presence of large sections which didn't fit with our linear model. All tests of the LTD in TRECVID have been performed with the single parameter T_P [15].

In Fig. 5 the overall results on both cuts and transitions are shown. The total number of participant groups was 21, but only the best 13 are visible. The LTD algorithm has obtained very good results in recall with a range of 88.5% to 91.4% of correctly detected transitions with the same algorithm for all different types of transitions. Since the LTD has no prevision for flashes, or picture in picture changes, the precision results are lower.

In Table IV the average results obtained by all TRECVID 2005 participants are reported. Overall results show that on cuts, LTD reached the best recall, while having 12^{th} position for precision; on gradual transitions LTD reached 7^{th} position for recall and 10^{th} position for precision. The frame recall (that is the number of transition frames recalled) is really low, because this algorithm leads to very high precision in the frame selection (only the linear part of the transition is identified).

V. CONCLUSION

We presented an algorithm for shot detection able to detect both abrupt cuts and gradual transitions in sports videos. Experimental results and comparisons show that the presented approach is clearly comparable with state-of-the-art algorithms

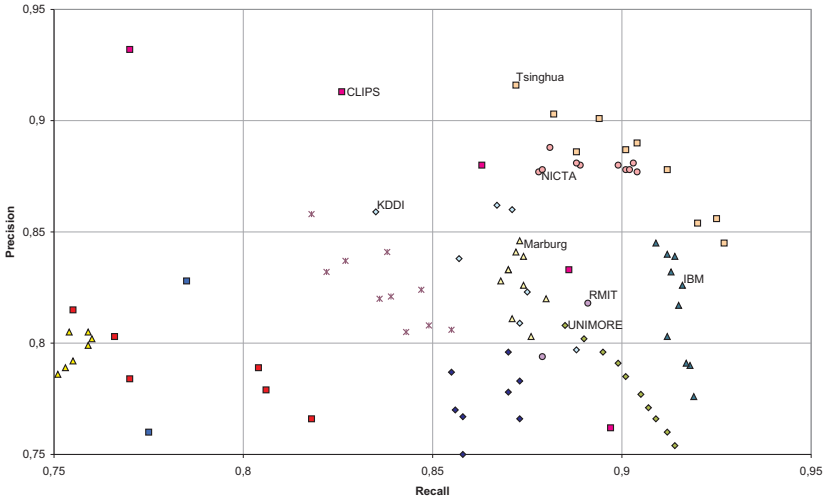


Fig. 5. Results on the overall detection (cuts and transitions) based on the data provided by the organizers. Our approach is labeled UNIMORE.

TABLE III

RESULTS COMPARISON FOR DIFFERENT ALGORITHMS IN THREE FORMULA 1 VIDEOS. WITH THE NOTATION "MORE RECALL" WE SELECTED A DIFFERENT WORKING POINT IN ORDER TO SHOW THAT THE RECALL MAY BE INCREASED, AT THE EXPENSES OF PRECISION.

	Formula 1 Italy						Formula 1 Austria						Formula 1 Europe					
	Cuts		**Trans**		**Overall**		**Cuts**		**Trans**		**Overall**		**Cuts**		**Trans**		**Overall**	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
VCM	92	90	66	38	93	74	91	92	39	34	91	83	87	95	26	13	87	72
VideoAnnex	46	84	28	89	55	86	50	86	16	97	54	88	45	86	26	95	54	88
LTD	84	95	67	84	88	91	86	96	50	87	88	95	93	95	81	84	94	92
LTD (more recall)	51	99	29	93	59	97	54	99	16	91	58	98	49	99	26	94	56	98

* Only dissolves were included in the analysis

TABLE IV

RESULTS OF TRECVID 2005 SHOT BOUNDARY DETECTION. LTD RESULTS ARE LABLED "S" AND MARKED IN BOLD.

	All		Cuts		Gradual			
							Frame	
	Recall	Prec	Recall	Prec	Recall	Prec	Recall	Prec
a	86.33	76.79	95.15	83.00	60.48	57.36	73.68	60.68
b	70.11	87.29	71.40	89.43	66.33	81.30	84.30	80.63
c	75.26	79.49	78.55	83.74	65.74	67.57	77.89	62.49
d	41.15	89.50	55.15	89.55	0.00	0.00	0.00	0.00
e	83.74	82.52	85.91	85.93	77.35	73.12	82.92	71.22
f	83.74	70.95	88.16	75.30	70.86	61.12	82.26	65.20
g	91.45	81.59	93.71	86.40	84.85	69.17	82.36	87.07
h	48.20	33.20	62.00	41.20	7.85	6.05	23.70	88.40
i	79.55	76.47	84.26	78.16	65.77	71.25	87.69	44.70
j	80.69	87.28	92.19	91.88	47.07	45.33	35.22	30.55
k	70.33	36.35	77.22	66.95	50.16	15.33	29.28	83.65
l	87.28	82.80	92.59	88.36	71.72	66.88	74.17	48.69
m	59.90	54.10	64.60	56.90	46.10	45.00	68.80	76.60
n	89.24	87.98	94.15	91.27	74.84	77.80	76.86	90.30
o	90.02	67.08	94.01	71.37	78.34	57.94	76.04	80.53
p	90.25	88.16	93.98	92.21	79.26	76.88	85.19	82.52
q	78.00	79.40	91.55	80.10	38.30	75.20	68.35	91.45
r	17.68	25.85	20.83	25.87	8.48	23.60	37.00	60.27
s	**90.17**	**78.10**	**95.72**	**82.32**	**73.96**	**65.46**	**65.36**	**89.43**
t	68.36	48.74	91.68	48.74	0.00	0.00	0.00	0.00
u	69.80	82.20	67.80	92.60	75.80	63.50	80.70	78.90

and outperformed a selected few on a specific test set. Moreover, it requires only two parameters, thus making the training process easier with respect to other approaches presented in the literature.

REFERENCES

[1] *Description of MPEG-7 Content Set*, ISO/IEC Std. JTC1/SC29/WG11/N2467, Oct. 1998. [Online]. Available: http://www.tnt.uni-hannover.de/project/mpeg/audio/public/mpeg7/w2467.html

[2] M. Naphade, R. Mehrotra, A. Ferman, J. Warnick, T. Huang, and A. Tekalp, "A high-performance shot boundary detection algorithm using multiple cues," in *Proceedings of International Conference on Image Processing*, 1998, pp. 884–887.

[3] S.-C. Pei and Y.-Z. Chou, "Efficient MPEG compressed video analysis using macroblock type information," *IEEE Trans. Multimedia*, vol. 1, no. 4, pp. 321–333, Dec. 1999.

[4] B.-L. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 533–544, Dec. 1995.

[5] W. Heng and K. Ngan, "Long transition analysis for digital video sequences," *Circuits, Systems and Signal Processing*, vol. 20, no. 2, pp. 113–141, 2001.

[6] C.-L. Huang and B.-Y. Liao, "A robust scene-change detection method for video segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1281–1288, Dec. 2001.

[7] J. Bescos, "Real-time shot change detection over online MPEG-2 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 475–484, 2004.

[8] J. Bescos, G. Cisneros, J. Martinez, J. Menendez, and J. Cabrera, "A unified model for techniques on video-shot transition detection," *IEEE Trans. Multimedia*, vol. 7, no. 4, pp. 293–307, Apr. 2005.

[9] R. Lienhart and A. Zaccarin, "A system for reliable dissolve detection in videos," in *Proceedings of International Conference on Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 406–409.

[10] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 2, pp. 90–105, Apr. 2002.

[11] B. Truong, C. Dorai, and S. Venkatesh, "New enhancements to cut, fade, and dissolve detection processes in video segmentation," in *Proceedings of ACM Multimedia*, Nov. 2002, pp. 219–227.

[12] A. Miene, A. Dammeyer, T. Hermes, and O. Herzog, "Advanced and adapted shot boundary detection," in *Proc. ECDL WS Generalized Documents*, 2001, pp. 39–43.
[13] VCM: Video content management by technologie-zentrum informatik (www.tzi.de). [Online]. Available: http://astral.ced.tuc.gr/delos/cls_resource_description.jsp?id=10129
[14] [Online]. Available: www.research.ibm.com/VideoAnnEx/
[15] Y. Zhai, J. Liu, X. Cao, A. Basharat, A. Hakeem, S. Ali, M. Shah, C. Grana, and R. Cucchiara, "Video understanding and content-based retrieval," in *TREC Video Retrieval Evaluation Workshop Online Proceedings (TRECVID2005)*, Gaithersburg, MD, USA, Nov. 2005. [Online]. Available: http://www-nlpir.nist.gov/projects/tvpubs/tv5.papers/ucf.pdf