MPEG-7 Compliant Shot Detection in Sport Videos

Costantino Grana, Giovanni Tardini, Rita Cucchiara
DII - Università degli Studi di Modena e Reggio Emilia
{grana.costantino, tardini.giovanni, cucchiara.rita}@unimore.it

Abstract

In this paper we propose a system for automatic detection of shots in sport videos. Our work covers two main aspects: the first is robust shot detection in presence of fast object motion and camera operations. To this aim we propose a new algorithm, unique for both cuts and linear transitions detection, which only needs the tuning of two parameters. An extended comparison with four transition detection algorithms, representing the state of the art in literature, is reported. Examples with Formula 1, Basket, Soccer and Cycling videos are analyzed. The second aspect is an in depth discussion on the annotation of shots and transitions with the MPEG-7 standard.

1. Introduction

Automatic tools for video segmentation and annotation are the chimera of all digital video library management systems. The goal is to find automatic and general procedures to segment videos into blocks and to annotate them with textual data or with metric information that could be useful for further indexing, querying, summarization, fast browsing and so on. Multimedia annotation tools need also standard output, compliant with other tools for browsing or indexing. MPEG-7 standard was defined to this purpose, by strictly indicating issues, descriptors and descriptors schemes useful for describing a video and part of it, even if it is still not used in commercial applications.

In this paper, we propose a new two steps iterative algorithm, which relies on a linear transition model, able to identify transition center and length. Its performance, are compared with state of the art approaches and are interesting in term of precision/recall and ease of tuning, since only two parameters are required. Finally, we discuss how to use MPEG-7 to create a standard and useful video annotation. This is not straightforward, since MPEG-7 recommendations are very general and different options can be chosen. Thus, in this paper we focus on some “good practices” to provide a fully compliant video segmentation description.

2. Related work

In recent years many techniques have been proposed for abrupt transitions (hence called *cut*) detection, and they have proved to give highly satisfactory results. The most common ones exploit differences of some metrics between adjacent frames [1]. Some of them have addressed the problem of shot detection in the compressed domain. The only information extracted from the videos in compressed domain approaches are those directly available from the MPEG streams, that is DCT coefficients, motion vectors and directions of prediction for each block [2]. While this kind of approach has achieved results comparable with the uncompressed domain techniques in cut detection, comparative studies have demonstrated that they perform much worse on gradual transitions [3]. Alternative approaches analyze frame windows, but this is difficult, since the variation between two different shots can be confused with the motion variation within the shot. For instance, the algorithm proposed in [4] tries to find a “plateau” in the difference values extracted with a single fixed frame-step. A refined approach is proposed by in [5], where authors deal with long transitions. Each frame is compared against a reference frame chosen from the sequence, and a one-dimensional changed indicator is computed. This indicator is claimed to be a ramp during a transition and constant in the same shot. They heuristically estimate the slope of the ramp and the standard deviation at the border to find transitions boundaries. A very similar method is proposed in [6], where each frame is compared against a fixed “seed”. When the window is long enough, difference values are uncorrelated, and a non decreasing ramp in values indicates a transition. A low pass filter is also exploited to detect and remove uncorrelated random noise.

In [7] a comparative study of most of the metrics used in shot detection approaches is presented, both in compressed and uncompressed domain. An algorithm is proposed, to detect both abrupt and gradual transitions. Here the author computes the frame differences using multiple frame-steps, and thus the final decision space is the union of multiple decision spaces, one for each frame step. Recently, the same author proposed a unified framework [8] for both cuts and transitions, which showed very good results. The drawback of this method is that 20 parameters are needed (10 for cuts and 10 for transitions) for the decision space, thus making the learning process more complex. A learning process is instead required in [9], where a probabilistic based algorithm is proposed to detect both abrupt and gradual transitions. After obtaining a priori likelihood functions by experiments, they take into account all the relevant knowledge to shot boundary detection, like shot-length distribution and visual discontinuity patterns at shot boundaries.

Another widely explored technique is the one proposed in [10], which address linear transition only. The propriety here exploited is that the mean and the variance of pixels' 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 allowing achieving more discriminative power than general techniques. We developed an iterative algorithm that, given a frame of possible transition, alternatively tries to find to best center position for the transition and the best length, by minimizing an error function, which measures the fitness of data to the linear model.

3. Video segmentation

Before describing our algorithm in detail, it is useful to define the ideal model of linear transition and to underline its important properties. These will be exploited by the algorithm to cope with non idealities and to measure the confidence of the detection.

3.1. The Transition Model

Let's consider two consecutive shots in a video sequence, the first one ending at frame *e*, and the second one starting at frame *s*, with $e < s$. If $s = e + 1$ we have an abrupt cut, otherwise there are some frames of gradual transitions between *e* and *s*.

To design a shot segmentation algorithm, two assumptions must be done: the first one is that a feature $F(t)$ is computable for each frame at time *t*, with the characteristic of being discriminating and almost constant within the shot; ideally

$$\begin{aligned} &F(t) = F(e), \forall t \le e \\ &F(t) = F(s), \forall t \ge s \\ &F(e) \ne F(s) \end{aligned} \tag{1}$$

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

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

Sometimes there is confusion on the definition of 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 includes in the definition of transition abrupt cuts too, as transitions 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 difference metric formulations, instead of computing the difference between the frames $F(i)$ and $F(i + w)$, with *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:

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

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 computed at half-frames. The reason for expressing the metric as $d\left[F(n-w), F(n+w)\right]$ instead of $d\left[F(n), F(n+2w)\right]$ will be explained in section 3.2.2.

In Fig. 1 we see an example of an ideal linear transition with $L = 5$, from a shot with white pixels to one with black pixels. 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

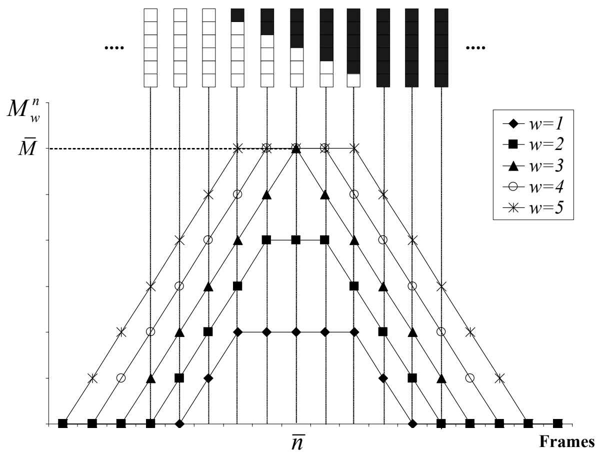

Figure 1. Values of M_w^n for an ideal linear transition with $L=5$ at varying *w*.

trapezoid centered in $\bar{n}$, for each *w*, that degenerates into a triangle when $2w = L+1$.

We can verify that in this ideal case, given the model and Eq. 3, both the up and down slopes last for $\min(2w, L+1)$ frames, and that the plateau of absolute maximum is $|2w-(L+1)|$ long. It's also straightforward to verify that:

$$M_w^{\bar{n}} < \bar{M}, \text{if } 2w < L+1; \qquad M_w^{\bar{n}} = \bar{M}, \text{if } 2w \geq L+1 \quad (4)$$

where $\bar{M} = \max_{w,n} M_w^n$ (see Fig. 1). We define $\psi_{w,L}^n(i;b,h)$ the generic trapezoidal function, centered in *n*, whose value is *h* at the center (the absolute height of the minor base) and *b* is the value outside the trapezoid. The function is plotted in Fig. 3. We define $\psi_{w,L}^n(i) = \psi_{w,L}^n(i;0,M_w^n)$, the function which corresponds to the ideal transition case.

In the real case, camera and objects motion, color and luminance variation and so on cause the feature *F* to be non constant on the shot, thus making Eq. 1 and Eq. 2 not satisfied. The consequence is that the shapes of both the slopes and the plateau are usually disturbed.

3.2. Two-steps Algorithm

Due to lack of ideality in most of the shot transitions, instead of relying only on correlation between data and the ideal $\psi_{w,L}^n(i)$ function, 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 searches for the transition length *L*, by trying different values of *w*, but keeping the transition center fixed. While in the ideal case even the first step would be sufficient, in real 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

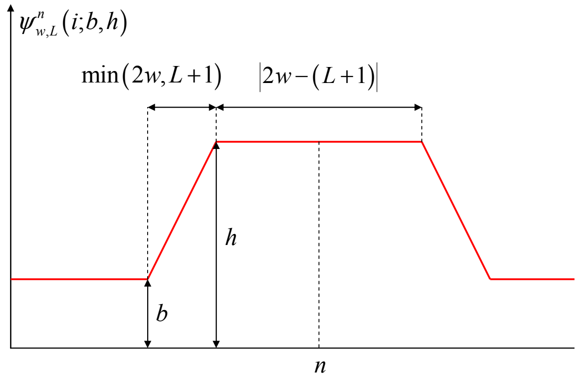

Figure 3. Trapezoidal shaped function $\psi_{w,L}^n(i;b,h)$

provide a different view of the function behavior, a possible confirmation on the first step outcome and a new estimate for the window size. Iteratively repeating the two steps allows progressively decreasing the error. In this section we explain in details our transition detection algorithm. We perform the following analysis on overlapped windows of 60 frames, distant 30 frames each other, since we suppose that transitions are much shorter and farther than that.

3.2.1. First step. In the first step the values of M_w^n are calculated using the frame-step $\bar{w}$, which is found in the previous iteration of the algorithm, or it's arbitrary chosen for the first iteration. The best trapezoid $\psi_{w,L}^n(i)$ is searched by moving the center *n*, and trying different values for *L*, but keeping $\bar{w}$ 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:

$$\Lambda_{\bar{w},L}^n = \sum_{i=n-\delta}^{n+\delta} \min\left(M_{\bar{w}}^i, \psi_{\bar{w},L}^n(i)\right) - \sum_{i=n-\delta}^{n+\delta} \left|M_{\bar{w}}^i - \psi_{\bar{w},L}^n(i)\right| \quad (5)$$

The value of *n* is searched within the 60 frames window, and also *L* must be selected such that $n+\delta$

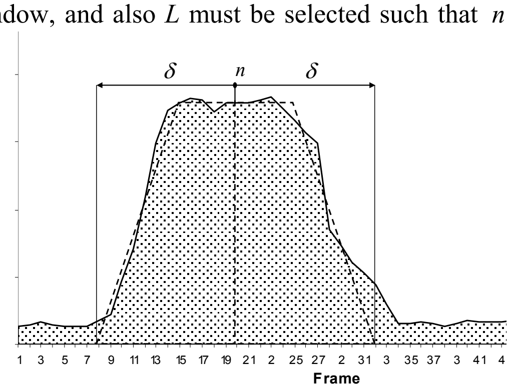

Figure 2. Example of real M_w^n values and the best trapezoid fitted.

and $n-\delta$ don't exceed the window.

In Eq. 5, two components are evident: the first one is needed to maximize the area 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 the distance measure to give a trapezoidal shape (the second term in Eq. 5), but we also request its *strength*, i.e. the amount of difference between the first and the second scene, to be significant. The first term in Eq. 5 in fact describes how much the value of M_w^n surpasses the ideal trapezoid. After finding the trapezoid which maximizes $\Lambda_{\bar{w},L}^n$, we consider $\bar{n} = \arg\max \Lambda_{\bar{w},L}^n$ the candidate transition center. In Fig. 2 we show an example of trapezoid fitting with real data.

3.2.2. Second Step. Thanks to the definition of M_w^n as a distance function centered in *n*, as in Eq. 3, increasing the frame-step *w* makes the value of M_w^n to grow up 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 can be plotted as shown in Fig. 4, with a linear slope followed by a horizontal line, when the value of M_w^n is stable. The second step of the algorithm uses this propriety to give an estimate of the transition length, by finding the smallest *w* which maximizes M_w^n. To provide a technique able to deal with noise, the tilt change of the chart is searched by minimizing the function:

$$Z_w^{\bar{n}} = \sum_{i=0}^{w} \left| M_i^{\bar{n}} - \frac{M_w^{\bar{n}}}{w} i \right| + \sum_{i=w+1}^{W} \left| M_i^{\bar{n}} - M_w^{\bar{n}} \right| \qquad (6)$$

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

In simple cases the algorithm progressively narrows the trapezoid minor base leading to the expected triangular shape. Convergence is not guaranteed in non ideal conditions, and, for this reason, we add a convergence constraint: at each iteration the minor base of $\psi_{w,L}^n(i)$ is forced to become smaller. In Fig. 5 the M_w^n 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.

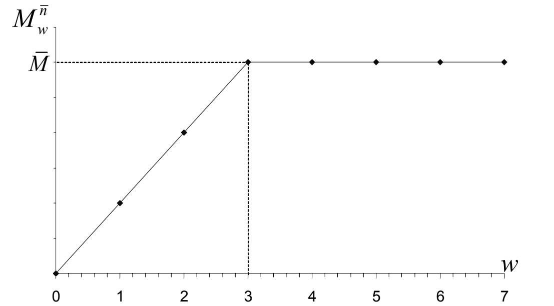

Fig. 4. Values of the distance metric M_w^n, with respect to different *w* values. This corresponds to the transition of Fig. 1.

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

$$Peak_{\bar{w}}^{\bar{n}} = M_{\bar{w}}^{\bar{n}} - \min\left(M_{\bar{w}}^{\bar{n}-2\bar{w}}, M_{\bar{w}}^{\bar{n}+2\bar{w}}\right). \qquad (7)$$

The Peak 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 $M_w^{n\pm 2w}=0$, 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 we have to get rid of the hypothesis of having an isosceles triangle and define the fitting error measure as:

$$err_{\bar{w}}^{\bar{n}} = \frac{1}{4\bar{w}} \sum_{i=1}^{2\bar{w}} \left| M_{\bar{w}}^{\bar{n}-i} - \psi_{\bar{w},L}^{\bar{n}}(\bar{n}-i, M_{\bar{w}}^{\bar{n}-2\bar{w}}, M_{\bar{w}}^{\bar{n}}) \right| + \left| M_{\bar{w}}^{\bar{n}+i} - \psi_{\bar{w},L}^{\bar{n}}(\bar{n}+i, M_{\bar{w}}^{\bar{n}+2\bar{w}}, M_{\bar{w}}^{\bar{n}}) \right| \qquad (8)$$

The error sum is divided by the triangle's base $4\bar{w}$ to obtain a measure which is independent from the transition length. A minimum threshold on the Peak value, T_P and a maximum threshold on error, T_E, are

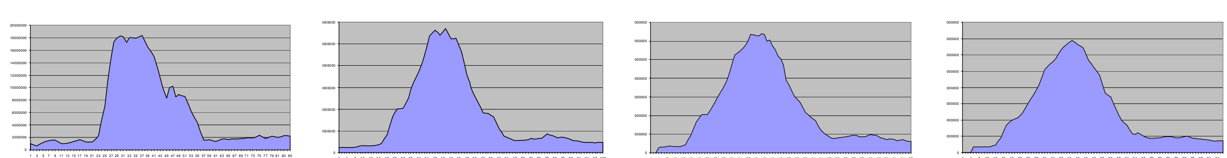
Figure 5. Four successive iterations of the algorithm in a real gradual transition: at each iteration, the shape of M_w^n values becomes more similar to a triangle

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001">
  xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001
  Mpeg7-2001.xsd">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AnalyticEditedVideoType">
    <AnalyticEditedVideo xsi:type="EditedVideoType">
    <MediaLocator xsi:type= "TemporalSegmentLocatorType">
      <MediaUri>MyEditedVideo.mpg</MediaUri>
    </MediaLocator>
    <AnalyticEditingTemporalDecomposition>
    <!-- Shots and Transitions -->
    </AnalyticEditingTemporalDecomposition>
    </AnalyticEditedVideo>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Figure 6. MPEG-7 description of a generic shot detection algorithm output

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 and transitions.

4. Shots description with MPEG-7

In order to index and reuse the outcome of shot segmentation a format to describe shot boundaries, the related transition boundaries and their characteristics must be defined. We didn't find in literature any universally accepted proposal for this, apart from the MPEG-7 description. Indeed it is worth noting that, to our knowledge, no publicly -commercial or non commercial- available software provides MPEG-7 compliant shot detection descriptions.

In clause 11.9 of part 5 of the standard (MDS) [11], the *AnalyticEditedVideoSegment* DS is introduced from which the specialized DSs *Shot* and *GlobalTransition* are derived. These two DSs can be employed alternating each other to describe the classical flow of an edited video. The complete XML tree of inclusions is shown in Fig.6. This schema can be easily replicated by every software, which provides segmentation. *Shots* are specialized *VideoSegments* that allow to include a *locationReliability* and an *editingLevelReliability*, plus further subdivision into compositions. From the example shown in Fig.7, a few considerations can be made. First of all, the MPEG-7 choice of showing as separate entities both shots and transitions allows the proper description of the characteristic of both entities, but forces the introduction of a virtual transition in case of cuts. In fact a cut is nothing but the abrupt join of two consecutive shots, so its temporal location is a conventional choice, which is never stated in the normative part and it is only shown in the examples. To be honest, the examples provided with MPEG-7 are contradictory, since the duration for a cut is set to 1, while in the ClassificationScheme it is clearly stated that the duration of a cut is 0. Another not very clear thing in MPEG-7 is the use of *TermUse* (for example in EvolutionType).

5. Results

For our tests 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 1, which also 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. 8 different types of correctly detected transition are shown. Row 1 shows an example of

```xml
<GlobalTransition evolutionReliability="0.775084">
  <MediaTime>
    <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F">71
      </MediaRelIncrTimePoint>
    <MediaIncrDuration mediaTimeUnit="PT1N25F">0
      </MediaIncrDuration>
  </MediaTime>
  <EvolutionType href="urn:mpeg7:cs:EvolutionTypeCS:2001:Cut"/>
</GlobalTransition>
<Shot id="VSEG01">
  <MediaLocator xsi:type="TemporalSegmentLocatorType">
    <BytePosition offset="916462"/>
  </MediaLocator>
  <MediaTime>
    <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F">71
      </MediaRelIncrTimePoint>
    <MediaIncrDuration mediaTimeUnit="PT1N25F">494
      </MediaIncrDuration>
  </MediaTime>
</Shot>
<GlobalTransition evolutionReliability="0.431947">
  <MediaTime>
    <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F">565
      </MediaRelIncrTimePoint>
    <MediaIncrDuration mediaTimeUnit="PT1N25F">3
      </MediaIncrDuration>
  </MediaTime>
  <EvolutionType href="urn:mpeg7:cs:EvolutionTypeCS:2001:Gradual"/>
</GlobalTransition>
```

Figure 7. MPEG-7 description of shots and transitions

Table 1. Video Set description and results in terms of precision (P) and recall (R) of our algorithm.

	# Frames	Abrupt Cuts	Gradual Transitions		Abrupt Cuts (%)		Gradual Transitions (%)		Overall (%)	
			Dissolves	Effects	P	R	P	R	P	R
F1 Italy*	124940	571	221	29	84	95	67	84	88	91
F1 Austria	138452	689	83	37	86	96	50	87	88	95
F1 Europe	153860	625	197	45	93	95	81	84	94	92
Soccer	22514	53	17	8	83	99	64	71	87	89
Basket	23361	75	26	12	96	83	88	56	97	74
Cycling	15407	2	43	0	17	100	78	84	79	84

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 correctly detected by our algorithm because they have a linear component included in their temporal extension (see figure caption for more details). In Fig 9 we plotted the values of M_w^n for the examples of Fig. 8, together with the triangular shaped function $\psi_{\bar{w},L}^{\bar{n}}$ used to calculate the error and the peak values. 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.

5.1. Algorithms Comparison

To compare the results of our algorithm we employed VCM [12] developed by Technologie-Zentrum Informatik of the University of Bremen and VideoAnnex [13] by IBM, which are freely downloadable shot detection software but with no source code available, and the algorithms described in [10] and in [7]. In Table 2 we show the comparison between our algorithm and the other ones for three Formula 1 videos. The training process, if needed, has been performed on the first video.

By looking at Table 2 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. Also its performance with abrupt cut is very poor, thus leading to mediocre overall results. On the contrary, the VCM algorithm is good on detecting cuts, but it has very poor results on gradual transitions.

In their paper, Truong et al. [10] proposed different algorithms for cut, fade and dissolve detection. We firstly implemented the dissolve detection algorithm. Since this algorithm is specifically designed to detected only dissolves, in the analysis we discarded every other type of gradual transitions. From the analysis of failure cases we can state that this algorithm can detect dissolves only when they appear in condition of no or very limited motion in the scene, that seldom happens in sports videos.

Concerning the Bescos algorithm, although we struggled to implement it exactly as described in [7], we must say that there is a big difference in the results declared in the paper and the ones we obtained. The comparison is possible since we used the same Cycling, Soccer and Basket videos extracted from the MPEG-7 Content Set. In out tests, the Bescos algorithm reached good results on cuts, while being rather weak on gradual transitions.

5.2. Good practices in MPEG-7 reporting

The only software which tries to provide some level of MPEG-7 output, is IBM VideoAnnex [13], which uses the *VideoSegment* Descriptor Schema (DS), to describe shots. Nevertheless, although compliant with the standard definition, this is not the correct way of reporting shots, since there is another more specialized descriptor for reporting shots and transitions details. Syntactical correctness doesn't imply semantically meaningful descriptions, and this is to our opinion one of the main problems with MPEG-7. In particular, if a specialization for a certain task is present in the standard, it should be definitely the tool of choice of software products. The main advantage of the description based on *AnaliticEditedVideo-Segment* over the basic generic class is that *Shots* are semantically distinguished by *GlobalTransitions*. Moreover GlobalTransitions allow describing the *EvolutionType* as a reference to a standard classification scheme (B.2.10) and the corresponding *evolutionReliability* can be used to show that we are not that sure of a particular transition. Attention should be paid to the fact that in case of cuts, the transion length must be 0, the first frame of the next shot is the one after the transition end, except in case

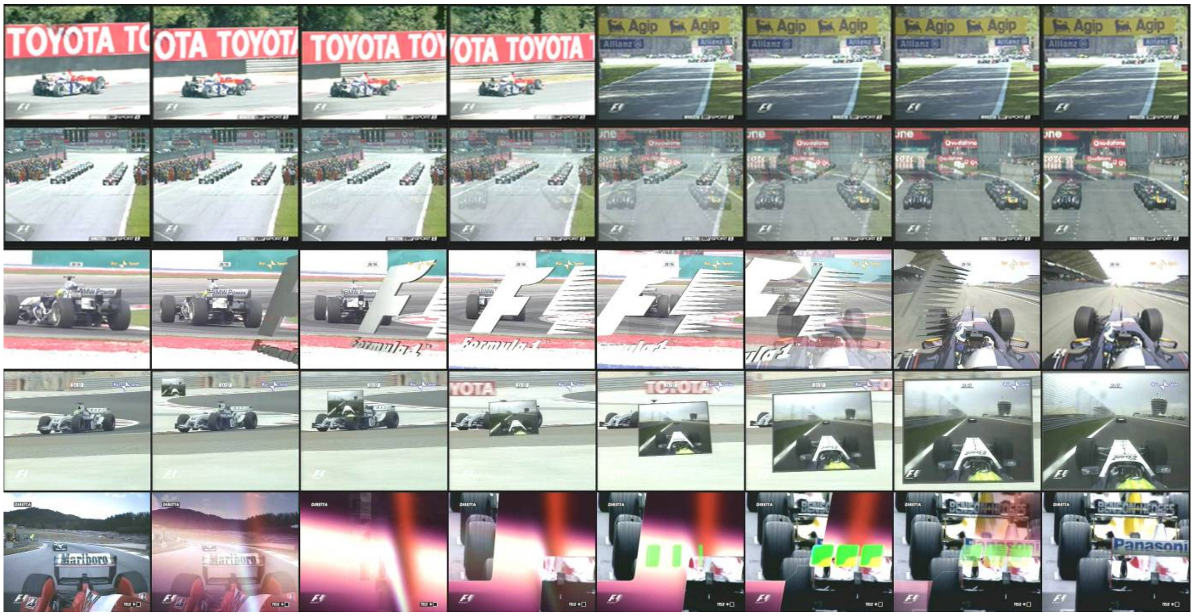

Figure 8. Examples of correctly detected transitions. From top to bottom: cut, dissolve, 3 special edit effects, which are correctly detected because they contain a linear part. For instance, in effect on row 3 there is a dissolve, which is clearly visible at the end. A dissolve is also present at the beginning of effect on the last row. The effect on row 4 shows a linear transform in its last frames.

of cuts, and even if cuts and gradual transitions can be distinguished by their length, it is sensible to use the EvolutionType. Instead, the use of the generic VideoSegment DS forces the software to make some assumptions on the generator of the MPEG-7 description intentions. For example VideoAnnex uses another *TemporalDecomposition* to include keyframes, but this is not the correct way of doing it, since the standard provides the *HierarchicalSummary* DS, which is devised to this aim.

6. Conclusions

We presented an algorithm for shot detection able to detect both abrupt cuts and gradual transitions in sport videos. Experimental results and comparisons show that our algorithm performs better than other techniques at the state of the art, and that it requires only two parameters, thus making the learning process easy. We have shown a complete MPEG-7 description, which can be used by every software performing shot detection, including all characteristics of the output. Most important, a few details not specified by the standard have been pointed out and possible solutions have been proposed. A demonstrator of the algorithm is available for download at *http://astral.ced.tuc.gr/delos/* under the *demonstrators* section.

7. Acknowledgments

The work is supported by the DELOS Network of Excellence on Digital Libraries, as part of the Information Society Technologies Program of the European Commission (Contract G038-507618).

8. References

[1] Gong, Y. An Accurate and Robust Method for Detecting Video Shot Boundaries. In *Proceedings of IEEE*

Table 2. Results comparison for different algorithms in 3 Formula 1 Videos

	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*
Truong*	-	-	*34*	*50*	-	-	-	-	*19*	*54*	-	-	-	-	*33*	*57*	-	-
Bescos	*80*	*93*	*51*	*61*	*84*	*84*	*64*	*95*	*14*	*50*	*85*	*88*	*83*	*88*	*51*	*46*	*86*	*76*
Our algorithm	*84*	*95*	*67*	*84*	*88*	*91*	*86*	*96*	*50*	*87*	*88*	*95*	*93*	*95*	*81*	*84*	*94*	*92*

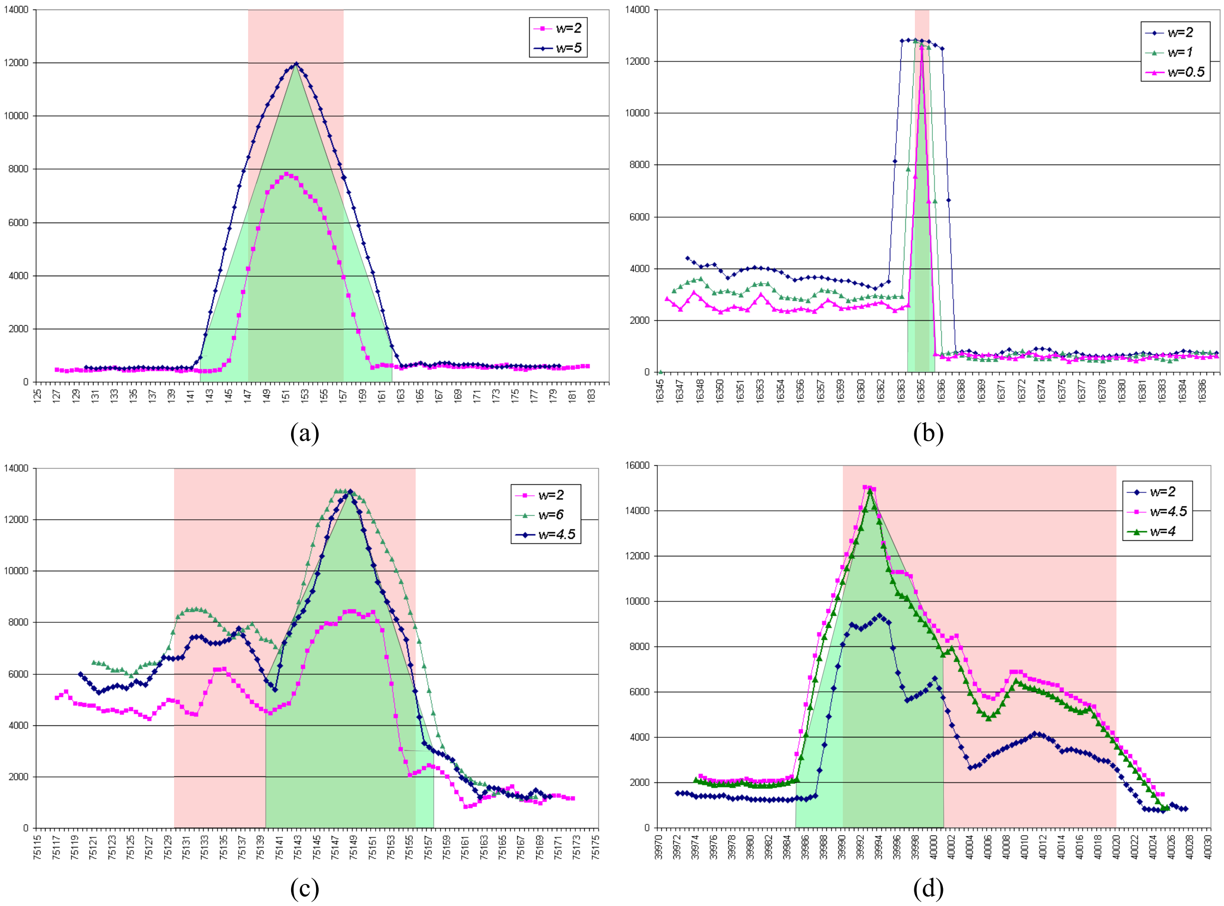

Figure 9. Values of M_w^n for different types of transitions. Every dot is plotted at half frame position. The rectangular shaded area represents the real extent of the transition. The triangular area is the one used by the algorithm to calculate the error and the peak value. (a) Cut, (b) Dissolve, (c) Special edit effect shown in row 3 of Fig 8, (d) Special edit effect show in row 5 of Fig. 8. Both these effects have a linear part (respectively at the end and at the beginning of the transition) that the algorithm correctly detects.

International Conference on Multimedia Computing and Systems, (Florence, Italy, June 07-11, 1999), 850-854.

[2] Pei, S.-C., Chou, Y.-Z. Efficient MPEG Compressed Video Analysis Using Macroblock Type Information. *IEEE Transactions on Multimedia*, 1, 4 (Dec. 1999), 321-333.

[3] Gargi, U., Kasturi, R., Strayer, S.H. Performance Characterization of Video-Shot-Change Detection Methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 10, 1 (Feb. 2000), 1-13.

[4] Yeo, B.-L., Liu, B. Rapid Scene Analysis on Compressed Video. *IEEE Trans. on Circuits and Systems for Video Technology*, 6 (Dec. 1995), 533-544.

[5] Heng, W.J., Ngan, K.N. Long transition analysis for digital video sequences. *Circuits, Systems and Signal Processing*, 20, 2 (2001), 113-141.

[6] Huang, C.-L., Liao, B.-Y. A robust scene-change detection method for video segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11, 12 (Dec. 2001), 1281-1288.

[7] Bescos, J. Real-Time Shot Change Detection Over Online MPEG-2 Video. *IEEE Trans. on Circuits and Systems for Video Technology*, 14, 4 (2004), 475-484.

[8] Bescos, J., Cisneros, G., Martinez, J.M., Menendez, J.M., Cabrera, J. A unified model for techniques on video-shot transition detection. *IEEE Transactions on Multimedia*, 7, 2 (Apr. 2005), 293-307.

[9] Hanjalic, A. Shot-boundary detection: unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12, 2 (Feb. 2002), 90-105.

[10] Truong, B.T., Dorai C., Venkatesh, S. New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *Proceedings of ACM Multimedia* (Nov. 2000), 219-227.

[11] Information technology - Multimedia content description interface - Part 5: Multimedia description schemes, ISO/IEC 15938-5:2003.

[12] VCM: Video Content Management by Technologie-Zentrum Informatik (www.tzi.de). Downloadable from http://astral.ced.tuc.gr/delos/cls_resource_description.jsp?id=10129

[13] www.research.ibm.com/VideoAnnEx/