Video Shot Boundary Detection: Seven Years of TRECVid Activity

Alan F. Smeaton [*]

CLARITY: Centre for Sensor Web Technologies
Dublin City University, Glasnevin, Dublin 9, Ireland

Paul Over

Information Access Division, National Institute of Standards and Technology,
Gaithersburg, Md., USA

Aiden R. Doherty

CLARITY: Centre for Sensor Web Technologies
Dublin City University, Glasnevin, Dublin 9, Ireland

Abstract

Shot boundary detection (SBD) is the process of automatically detecting the boundaries between shots in video. It is a problem which has attracted much attention since video became available in digital form as it is an essential pre-processing step to almost all video analysis, indexing, summarisation, search, and other content-based operations. Automatic SBD was one of the *tracks* of activity within the annual TRECVid benchmarking exercise, each year from 2001 to 2007 inclusive. Over those seven years we have seen 57 different research groups from across the world work to determine the best approaches to SBD while using a common dataset and common scoring metrics. In this paper we present an overview of the TRECVid shot boundary detection task, a high-level overview of the most significant of the approaches taken, and a comparison of performances, focussing on one year (2005) as an example.

Key words: Shot boundary detection, TRECVid, evaluation

[*] Corresponding author.
Email addresses: `Alan.Smeaton@DCU.ie` (Alan F. Smeaton), `over@nist,gov` (Paul Over), `adoherty@computing.dcu.ie` (Aiden R. Doherty).

1 Introduction

Automatic shot boundary detection (SBD) is an enabling function for almost all automatic structuring of video. It has been the subject of much investigation over many years and a large variety of techniques have been proposed, and evaluated, from the very simple comparison of adjacent frames, to the more complex recognition of patterns of motion vectors in compressed video. Extensive reviews of the techniques which can and have been used for SBD are beyond the scope of this paper and we refer the reader to [1,2,3,4,5,6] for further information on these.

One of the downsides of most the previous research which reports investigation into shot boundary detection is that for the most part, any evaluation of performance tends to be on small, unique collections of video. Few places in the literature report comparisons of techniques on the same video, and where they do, the number and variety of techniques included in such direct comparisons is very small, for example [7]. An exception to this is TRECVid.

TRECVid is a large-scale, worldwide benchmarking activity running annually, whose goal is to encourage research into tasks related to content-based information retrieval on digital video. It does this by providing a large video test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results [8]. Between 2001 and 2007, TRECVid supported evaluation of the task of shot boundary detection where a large variety of SBD techniques from 57 different research groups worldwide were benchmarked each year on the same video using the same scoring mechanisms and with the same manually created groundtruth. In this paper we present an overview of the TRECVid SBD task and we examine its achievements. We give a high-level overview of the most significant of the approaches taken to SBD, and we present a comparison of performances, focussing on one year (2005) as an example. There are several reasons why we focus on 2005 including the fact that it is the year of the largest of the video collections, but mostly it is because 2005 was the first year where we introduced multiple TV channel sources, and languages, into the test collection and this offered the greatest challenge to participants with no chance of their detection techniques being attunded to any one TV source.

The rest of this paper is organised as follows. In the next section we describe the TRECVid SBD evaluation process, the data which was used each year and which varied from year to year, and the scoring mechanisms and evaluation metrics which were developed in the early years and then used subsequently. Section 3, the largest section, provides a categorisation of some of the most significant techniques used in SBD, concentrating on 10 of the approaches taken in TRECVid in 2005. Section 4 presents a comparison of the performances of the techniques described. A concluding section provides a re-cap of our view of the state of the art in SBD and some indications of where we believe the research challenges remain.

2 The TRECVid Evaluation Process

Since its inception as a track within the TREC benchmarking in 2001, TRECVid has followed the same operational model. This involves gathering video data and distributing it to participating groups, allowing groups to run their techniques on this test data and to submit the results of their experiments back to the coordinator, NIST (the National Institute of Standards and Technology in Gaithersburg, Md., USA), before some deadline. NIST then pool all the submitted results, eliminate duplicates from across participants and then manually assess these pooled results for accuracy. Each year in which the SBD task was run as part of TRECVid, test video data was provided by NIST (usually on DVD) to participating researchers a few weeks before the output of the individual group systems run on this test data (submissions) was due. The SBD submissions were evaluated automatically using software created at NIST in 2002 and then made publicly available on the TRECVid website. The evaluation software compared the submitted results for SBD to the groundtruth manually produced by the NIST annotator. Detailed and summary performance figures were then returned to the participants for analysis.

2.1 Test data

The test data for each year have been a representative, usually random, sample of approximately 6 hours of the video used in the main TRECVid tasks such as search and feature detection. The origins and genre types of the video data have varied widely from the initial NIST and NASA science videos in 2001, to the Prelinger Archive's antique, ephemeral video, to broadcast news from major US networks in the mid-1990's to more recent Arabic and Chinese TV news programming. Editing styles have changed and with them the shot size and distribution of shot transitions types. Some of these characteristics are listed in Table 1. The Sound and Vision data used in 2007 stands out for the longer shots and lack of gradual transitions compared with the other sources and this is because of the nature of education, news magazine and historical TV compared to broadcast TV news.

Table 1
Shot boundary detection test data

Year	Hrs.	Files	Frames	Trans	%Cut	%Diss.	%Fade	%Other	Data description
2001	5.8	42	594,170	3,176	65.0	30.7	1.7	2.6	Open-Video Project, NIST videos, BBC stock shots
2002	4.8	18	545,068	2,090	70.1	24.4	3.0	2.4	Prelinger Archive, Open-video
2003	6	13	596,054	3,734	70.7	20.2	3.1	5.9	English broadcast TV news (ABC & CNN)
2004	6	12	618,409	4,806	57.7	31.7	4.8	5.7	English broadcast TV news (ABC & CNN)
2005	7	12	744,604	4,535	60.8	30.5	1.8	6.9	Arabic (LBC), Chinese (CCTV-4 & NTDTV), English (CNN, NBC, & MSNBC) broadcast TV news
2006	7.5	13	597,043	3,765	48.7	39.9	1.3	10.1	Arabic (LBC & ALH), Chinese (CCTV4, PHOENIX, & NTDTV), English broadcast TV news (NBC, CNN, & MSN)
2007	6	17	637,805	2,317	90.8	5.4	1.0	3.7	Sound & Vision educational, news magazine, historical

2.2 Groundtruth

The groundtruth for shot bounds was created by a researcher at NIST using the publicly available tool, VirtualDub [9], to examine each of the test videos, identify each shot transition, and label it as to type, whether it is a hard cut, dissolve, fade to/from black, or other. A script was used to sanity check the annotation and difficult cases were discussed with the TRECVid project leader before resolving. Analysis of annotator variation in determining groundtruth was not possible because we aimed to maintain consistency by using the same annotator to evaluate the SBD task across all seven years.

While categorization of most transitions posed no problems, there were some that did. For example, video content was sometimes presented as a picture-within-picture, with changes happening both inside the inner picture and outside. More complicated still were situations in which there were multiple windows within each frame, sometimes running independently of each other. In these cases an attempt was made to judge based on the most salient (larger, more central, etc.) part of the frame area.

2.3 Evaluation measures

Each year, participating groups in the SBD task were allowed up to 10 independent submissions or variations of their own approaches and these were compared automatically to the shot boundary reference data. Each group determined different parameter settings for each run they submitted and many explored precision/recall tradeoffs.

Submissions were compared to the shot boundary reference data using a modified version of the protocol proposed for the OT10.3 Thematic Operation (Evaluation and Comparison of Video Shot Segmentation Methods) of the GT10 Working Group (Multimedia Indexing) of the ISIS Coordinated Research Project [10].

For continuity with earlier work, the following measures were calculated by the NIST software: inserted transition count, deleted transition count, correction rate, deletion rate, insertion rate, error rate, quality index, correction probability, recall, and precision. The interested reader should see [10] for details on the definitions of these. Precision and recall were the primary measures used in the presentation of results at TRECVid. Where a single value for detection was needed, recall (R) and precision (P) were combined with equal weights in the F-measure (2*R*P/(R+P)).

Detection performance for cuts and for gradual transitions were both measured

Fig. 1. Overview of how cuts are evaluated

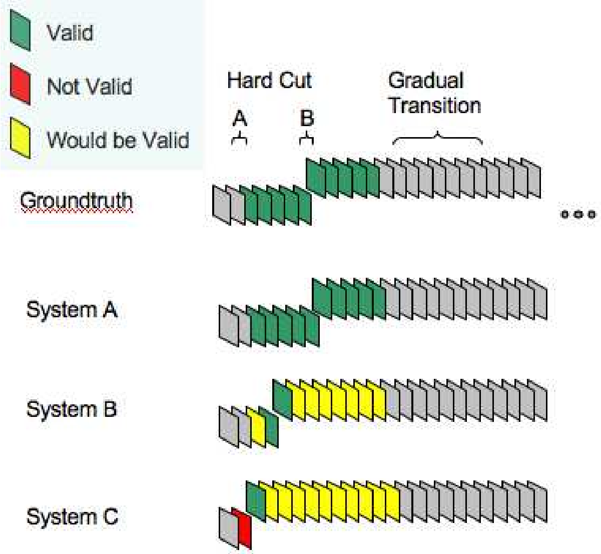


Table 2
Short graduals (1-5 frames)

	2003	2004	2005	2006
% of all transitions	2	10	14	24
% of all graduals	7	24	35	47
% of SG's = 1 frame	41	88	83	82

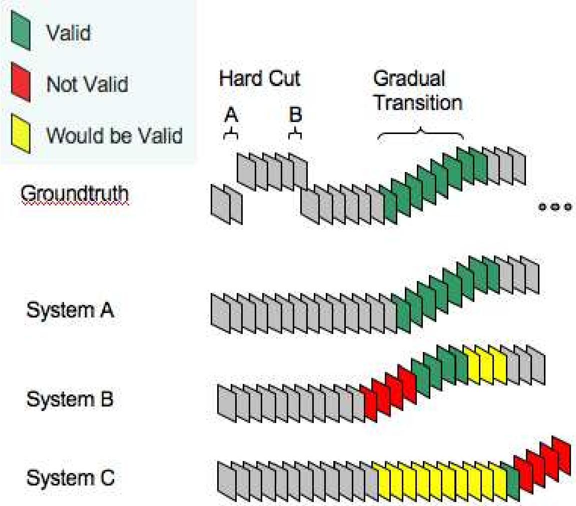
by precision and recall where the detection criteria (after 2001) required only a single frame overlap between the submitted transitions and the reference transition. This was to make the detection independent of the accuracy of the detected boundaries. For the purposes of detection, a submitted abrupt transition was considered to include the last pre-transition and first post-transition frames so that it has an effective length of two frames (rather than zero).

Gradual transitions could only match gradual transitions and cuts match only cuts, except in the case of very short gradual transitions (5 frames or less), which, whether in the reference set or in a submission, were treated as cuts. Each abrupt reference transition was expanded by 5 frames in each direction before matching against submitted transitions to accommodate differences in frame numbering by different decoders (see green frames around hard cut in Figure 1.

The notion of treating short gradual transitions as cuts was carried over from the ISIS project and took on an unexpected importance with the increase in numbers of short graduals over the years as shown in Table 2.

Accuracy for reference gradual transitions, which were successfully detected,

Fig. 2. Overview of how gradual transitions are evaluated



was measured using the one-to-one matching list output by the detection evaluation. The accuracy measures were frame-based precision and frame-based recall, new measures developed specifically within the context of TRECVid to measure the performance of detecting gradual shot transitions. These measures evaluated the performance of gradual shot transitions in terms of the numbers of frames overlapping in the identified, and the submitted gradual transitions, and thus higher performance using these is more difficult to achieve than for non- frame precision and recall. Note that a system could be very good in detection and have poor accuracy (high recall, low precision e.g. few yellow frames but many red frames in Figure 2), or it might miss a lot of transitions but still be very accurate on the ones it finds (high precision, low recall e.g. few red frames but many yellow frames in Figure 2).

In the next section we will present an overview of the main SBD techniques used by TRECVid participants.

3 TRECVid SBD Techniques

Throughout the 7-year history of the SBD task in TRECVid, 57 different research groups participated and completed submissions at least once[1] . With 109 runs over the 7 years from 57 participants, each representing a different approach to SBD with up to 10 experimental variations for each, this represents a very large diversity of experimentation. In this section we attempt to analyse some of these submissions in order to identify trends and commonalities, focusing on 2005 as an example year in order to limit the scope.

[1] http://www.computing.dcu.ie/~adoherty/sbd_review/participation_table.htm lists the participants from each year

Table 3
Approaches taken by participating groups (TRECVid 2005)

Rank	Group	MLrn	ColHst [2]	Flash	LVals	Cmpr	AThr	MCmp	Edgs	STmp	Other
1	Tsinghua University	✔	✔^{48}	✔	-	-	-	✔	-	-	-
2	National ICT Australia (NICTA)	✔	-	-	-	-	-	-	-	-	-
3	IBM Research	✔	✔^{512}	✔	✔	-	✔	-	✔	-	-
4	CLIPS-IMAG	-	-	✔	-	-	✔	✔	-	-	-
5	KDDI R&D Labs Inc.	✔	✔	✔	✔	✔	✔	✔	✔	-	-
6	University of Marburg	✔	✔^{512}	✔	✔	✔	-	✔	✔	-	-
7	RMIT University	✔	✔^{16}	-	-	-	✔	-	-	-	-
8	U. Central Florida & U. Modena	✔	✔	-	✔	✔	-	-	-	-	-
9	FX Palo Alto Laboratory (FXPal)	✔	✔	-	-	-	-	-	-	-	-
10	City Univ. Hong Kong	✔	✔^{512}	✔	✔	✔	-	-	✔	✔	-
11	Technical Univ. Delft	-	-	✔	✔	-	-	-	-	✔	-
12	Imperial College London	-	✔	✔	-	-	-	-	-	-	-
13	Hong Kong Polytechnic Univ.	-	✔	✔	-	-	-	-	-	-	✔
14	Fudan University	✔	✔	-	✔	✔	✔	-	-	-	-
15	University of Sao Paulo	-	✔	-	-	-	-	-	-	-	-
16	LaBRI	-	✔	-	✔	✔	-	✔	-	-	
17	Motorola Multimedia Res. Lab.	-	-	-	-	-	-	-	-	-	✔
18	Indian Institute of Technology (IIT)	-	-	✔	-	-	-	-	-	-	✔
19	University of Iowa	-	✔^{16}	-	-	-	-	-	✔	-	-
20	University Rey Juan Carlos	-	✔^{16}	✔	-	-	✔	-	-	-	✔
21	Florida International Univ.	-	-	-	-	-	-	-	-	-	✔

Table 3 shows a breakdown of what we consider the major approaches from participants in TRECVid 2005, ranked by their overall performance, where the column headings refer to approaches using machine learning (MLrn), those using colour histograms (ColHst), those with specific detectors for photo flashes (Flash), those where luminance values were used in frame comparison (LVals), those which operated only in the compressed (MPEG-1) domain (Cmpr), those incorporating adaptive thresholding (AThr), or motion compensation (MCmp), those which detected and used edges (Edgs) or other spatio-temporal characteristics (STmp), and a miscellany of other techniques (Other). We will now expand on some of these.

3.1 Execution speed and working in the compressed domain

Some measure of system effort was added to the measures to highlight limitations and improvments in algorithm speed. Hardware tended to be very similar and participants were asked to submit timing information for each of their system runs.

The first consideration we examined is those groups who decided to work only on the encoded MPEG-1 video data. As Table 3 shows, only 6 of the 21 groups did this, however it is significant that 3 of these groups were the 3 fastest overall in terms of execution time.Two of these groups are highlighted in bold font in Table 4 which shows the overall F-score and approximately normalised execution time for detection for the top 10 performing groups in TRECVid 2005. Almost all groups used a standard PC as their platform. Decoding video is often the most time-consuming part of any kind of video analysis so any way to reduce this time, by not decoding it at all for example, leads to faster processing.

It is interesting though that groups can be so much faster than real-time while working on uncompressed MPEG data. In the first SBD task in 2001, Microsoft Research Asia [11] found that working on the uncompressed domain can be very fast, operating at 1.5 times real-time on a Pentium-3 450MHz machine. With 4 years of further developments in the field to 2005, as well as working on faster processors, it is no surprise to find that groups like KDDI, U. Marburg and Tsinghua U. can operate even more quickly. The "Time" column of Table 4 shows the execution time taken for each of the top 10 performing groups, relative to real time. We can see that the U. Marburg, taking only 12% of real time, is the fastest and ranked 6th in accuracy, while the average time taken across these top 10 is about real time if we exclude the 10th ranked, which is a bit of an outlier in terms of execution speed.

Table 4
Top 10 performing groups from TRECVid 2005 (overall) and their execution speed

Rank	Group	F	Time	Processing
1.	Tsinghua U.	0.897	x 0.23	P4 3GHz
2.	NICTA	0.892	x 2.30	P4 3GHz
3.	IBM Research	0.876	x 0.30	x86 Family 15 Mdl 2 3.1GHz
4.	CLIPS-IMAG	0.876	x 2.30	P4 3.2 GHz
5.	**KDDI R&D**	**0.865**	**x 0.14**	P4 1.8/3.2GHz
6.	**Marburg U.**	**0.859**	**x 0.12**	P4 3GHz
7.	RMIT U.	0.853	x 0.89	AMD Athlon-64 3400+
8.	U. Modena	0.845	x 1.48	AMD Athlon XP2000+ 1.68GHz
9.	FX Palo Alto Lab.	0.839	x 1.77	AMD 64 3500
10.	City U. Hong Kong	0.831	x 16.0	P4 3GHz

3.2 Comparing frames to identify shot boundaries

It is important to establish how frames should be compared to each other to find out if a shot boundary has occurred. There are 3 main approaches to this, namely Colour Histograms (used by 15 groups), Luminance values (used by 8 groups), and Edges (used by 5 groups). Many groups use a combination of these techniques, and in fact 4 groups used all three approaches in 2005 and it is noticeable that they rank quite high in terms of F value[3] : IBM Research (ranked 3rd), KDDI (5th), Marburg (6th), and City University of Hong Kong (10th).

Other approaches to frame comparison considered by some groups include a comparison of frame thumbnail based gray-levels, a black frame detector, a monochrome frame detector, and a non-linear state-based fusion of techniques (all IBM); a computation of global difference between frames with gain and offset compensation (CLIPS); using image texture (Hong Kong); using image shape (U. Rey Juan Carlos); and making use of wavelets (IIT).

3.2.1 Using color histograms

The colour histogram approach is based on computing the colour of every pixel in each frame and gathering these into a histogram with a fixed number of "bins". The histograms of successive frames are compared to each other and

[3] National ICT Australia (ranked 2nd) use visual features, but do not explicitly state which ones

if they vary significantly then it is probable that a cut or gradual transition has occurred. This is an old technique which has been used for over a decade [3],

Many groups make use of colour histograms in TRECVid 2005 and the average number of colour bins used appears to be 512. Groups using this number of bins include IBM Research, U. Marburg, Hong Kong Polytechnic U., while U. of Iowa and University of Rey Juan Carlos use 16 quantizied bins. The other groups using colour histograms did not state how many bins they use and they include Tsinghua U., KDDI, FXPal, City University of Hong Kong, Fudan U., U. Sao Paulo, and LaBRI. RMIT University tried using 2 histogram types. One was a normal localised HSV histogram with 16 regions, while the other was a 3 dimensional global HSV histogram where each colour is represented as a point in 3D space, with 16 bins per colour component. In general though a higher number of bins in 1 dimensional colour histograms have been the preferred approaches of participants.

In terms of comparing color histograms a variety of methods are used by the top performing participants including: Manhattan distance (RMIT & Tsinghua), Histogram Intersection (Marburg), X^2 (U. Central Florida), and Chi-Squared (FX Palo Alto Laboratory). However many participants did not mention the distance measures they used, indicating the general feeling that choice of distance metrics is not so influential in affecting the final system performance. Indeed research in similar domains indicates that the simple Manhattan and Euclidean distance metrics are highly effective [12,13].

3.2.2 Using luminance values

Some groups compared the luminance values of different frames. In total, 8 groups made use of this feature, and used it in conjunction with either/both of colour histograms and edges. The groups making use of this feature were IBM Research, KDDI, U. Marburg, U. Florida/Modena, Hong Kong, Delft, and Fudan.

3.2.3 Using edges

Although colour histograms are still the most popular feature to help determine differences between frames, another approach that can be used, and even in conjunction with colour histograms, is to look for edges in each frame. If the edges from successive frames differ significantly then there's a strong possibility that a shot boundary has occurred.

The team from IBM Research used a 3D localised edge histogram, where the frame is divided up into 8x8 blocks and a total of 512 bins. KDDI took edges into account for determining fade/dissolve transitions, whereby they extracted two edge features from the Edge Histogram Descriptor specified in MPEG-7. The University of Marburg stated that they make use of *"edge histograms of Sobel-filtered (vertically and horizontally) DC-frames"*, while University of Iowa used aggregated edge distance, and City University of Hong Kong in previous work mention their use of edge features too.

While none of the top performing groups used edge detection alone, it is interesting to note that 3 of the top 6 groups used this in combination with colour histograms. It is worthwhile to note that KDDI, using the standard MPEG-7 edge histogram are among those 3 groups.

3.2.4 Other approaches

There were a variety of other, miscellaneous approaches which appeared among the participants' techniques. For example, Hong Kong Polytechnic Univ. used a distance map where a shot cut will appear as a triangle, a photo flash as 2 straight lines, and a gradual transition as a trapezoid. Based on the visual appearance of the distance map, the different transitions can be detected and classified easily [14].

Indian IT believe that the Morlet wavelet [15] gave a good discrimination between actual shots and false positives.

University Rey Juan Carlos made use of shape features to help find shot boundaries. If a shape is extracted from one frame and is not recognised in another succeeding frame, then there is a high probability that a shot boundary has occured.

The City University of Hong Kong compared frames based on their textures. They used this in conjunction with colour histograms, edge, and luminance values. As with other approaches, if the texture of 2 frames is significantly different then there is a chance that a shot boundary has occurred.

3.3 Finding gradual transitions

Hard cut detection is quite straigtforward in that it can be accomplished effectively when neighbouring frames are compared. However a gradual transition is more arbitrary in terms of how many frames to take into account. The average gradual transition is around 10 frames or so for the video data used in TRECVid.

The following groups used different numbers of neighbouring frames to construct average group/block values to take into account while trying to detect gradual transitions: CLIPS (5 frames), RMIT (14 frames), FXPal (10 frames), Hong Kong (>15 frames), Imperial College (16 frames), Hong Kong Polytechnic (10 frames), and Fudan (10-12 frames). These groups computed frame-frame similarity across these larger ranges and were thus able to detect gradual transitions. Other groups did not state their window sizes, but it can be seen that a window size of approximately 10 frames is most commonly chosen.

There are other interesting approaches to gradual transition detection. IBM Research used the same method for cut and gradual transition detection. They make use of a graph-based, multiple pair-wise frames comparison method. Each frame is a node in a graph. Pairs of frames, up to 13 frames apart, are connected with arcs and a shot transition appears as a cut in the graph [16]. The LaBRI group worked very much on the encoded MPEG-1 data and compared neighbouring I-frames in order to determine if a shot boundary has occurred. I-frames are the frames of video which MPEG-1 encodes in their entirety, independently of neighbouring video frames.

3.4 Machine learning

Machine learning operates by taking in sample data to help train a machine as to what cuts or gradual transitions appear like. This then helps the machine determine when cuts or gradual transitions occur in new data. Many groups (Tsinghua, ICT Australia, KDDI, Hong Kong, Fudan) make use of support vector machine classifiers (SVMs) to help detect either cuts or gradual transitions, or both. This is probably because of the ease of use of off-the-shelf machine learning tools which are now readily available. For training classifiers, almost all groups used image features, in particular colour taken from different colour spaces, as the features to be learned for classification and taken from each decoded frame in the video. Extracting colour features is fast compared to other image features such as, say edges, and this is one of the attractions of using colour and colour bins as features for machine learning. When using colour features, regional colours, with frames divided into $n \times m$ grids rather than global features exclusively, were common. Unsurprisingly, other video features such as optical flow or motion levels, did not feature in the machine learning techniques used

Using SVMs for classification were not the only uses of machine learning. University of Marburg use an unsupervised k-means clustering for both cut and gradual transition detection using 2 classifiers: adaboost-based and SVM; while FX Palo Alto Laboratory used a k-Nearest-Neighbour classifier to label each frame as either a shot boundary or non-boundary.

However we believe that the main take home message regarding classifiers is that 7 of the top 10 performing groups made use of SVMs. This provides a strong indication that there exists a belief in the community that utilising SVMs is well suited towards the task of identifying shot boundaries in video. Indeed once trained the execution speed of the classifiers is very good, as has been detailed in Table 4.

3.5 Flash detection

Photographic flashes can occur in video, especially TV news video, and can trigger the false detection of shot bounds. Several groups specifically targetted reducing the number of shot boundaries wrongly detected due to a camera flashes, or a flash of lightning, or a light being switched on in a scene. Most groups relied on a post-processing step where they try to remove "false alarms" after all other processing. The general approach taken towards this was to compare roughly 2 frames previous to the current frame, against 2 frames following the current frame. If there is no signigicant difference between these frames, it is presumed that a flash occurred and therefore this will not be recorded as a shot boundary.

3.6 Adaptive thresholds

In order for participants not using SVMs to quantify the difference between frames, there must be a threshold used so as to determine if the frame in question is a shot boundary or not. Adaptive threshold values change depending on the circumstances around the particular frame due to, for example, a change in the genre of the video.

Both CLIPS and RMIT University used a noise factor to dynamically adjust the threshold for detecting if a shot boundary should be triggered or not. Meanwhile IBM Research and U. Rey Juan Carlos considered frames either side of the frame in question to make an individual threshold for that frame. KDDI stated that they use a luminance adaptive threshold, which appears to be similar to the IBM Research and U. Rey Juan Carlos approach.

3.7 Motion compensation

In many scenes it is not unusual for a person or camera to move. We don't want to record this as a break in the scene or shot, therefore some groups specifically targetted motion compensation to reduce the chance of false triggers. CLIPS performed this technique using an optical flow approach which was able to align both images over an intermediate one. KDDI applied motion compensation through using reduced size motion vectors in encoded MPEG streams while U. Marburg used motion-compensated pixel differences of DC-frames.

3.8 Spatio-temporal slices

Spatio-temporal slices are 2D images extracted from videos with 1 dimension in space and the other in time. With spatio-temporal slices, less of the frame area is taken into account, thus processing costs are reduced, as reported by the City University of Hong Kong.

Hong Kong Polytechnic University made use of slice coherence for cut and wipe detection, and for dissolve and non-dissolve classification. They consider fade-in and fade-out as special cases of dissolve. The Technical University of Delft extracted their features from spatiotemporal video data blocks, so as to provide elementary evidence on the presence of a shot transition in the observed time interval.

3.9 Summing Up

Table 5, derived from the data in Table 3, details how popular some of the techniques were among the top-10 performing groups out of the 21 participants in TRECVid 2005. The "Top 10" column in particular provides an indication of the more successful techniques that should be included in a SBD system. From this we can see that machine learning is core to most approaches and that using colour histograms remains a fundamental component of the most successful approaches. Other techniques such as detecting camera flashes or working only in the compressed domain, are niche and not yet of widespread applicability.

4 Gauging the changing difficulty of the test data

One of the TRECVid participating groups, the Laboratoire d'Informatique de Grenoble (LIG), formerly CLIPS-IMAG, submitted the output of their same shot boundary detection system in each year of TRECVid's shot boundary task and have consistently placed among the top systems [17]. The developers report that the system did not significantly change since 2003, so results for this system provide evidence about the relative difficulty of the data as

Table 5
Popularity of Approaches among top 10 (of 21) systems in TRECVid 2005

Approach	Top 10	Total (of 21)
Machine Learning	9	10
Colour Histograms	8	15
Flash Detection	6	11
Luminance Values	5	8
Compressed Domain	4	6
Adaptive Thresholds	4	6
Motion Compensation	4	5
Edges	4	5
Spatio-temporal Slices	1	2
Other Approaches	0	3

we moved from year to year. Figure 3 shows the scores for the CLIPS/LIG system on cuts across the test collections from 2003 to 2007. Figure 4 shows corresponding scores for the gradual transitions. Results against gradual transitions for 2007 have been omitted because of the relatively small number of such transitions (227 out of a total of 2317 transitions), only about one quarter of the fraction in earlier years' test data. In addition to these figures for all (10) runs submitted by this group, the F-measure (the harmonic mean, combining precision and recall with equal weight) for the best CLIPS/LIG run for each year is detailed in Table 6

The results from this comparison across the years shows that the 2007 Sound and Vision hard cuts are easiest of all years. For both cuts and gradual transitions the dataset for 2003 is easier than that for 2004, though the difference is less for the gradual transitions. The dataset for 2006 contained the hardest cuts from among all years and especially the hardest gradual transitions. These results can be taken into consideration and used as a loose normalisation factor if comparing the performance of different approaches across different years.

5 Performances of SBD Techniques

5.1 Raw comparative performances

By focusing on shot boundary detection results submitted within the same year we can compare these results against each other directly. Tables 7 and 8

Fig. 3. Varying levels of difficulty of TRECVid cuts, 2003-2007

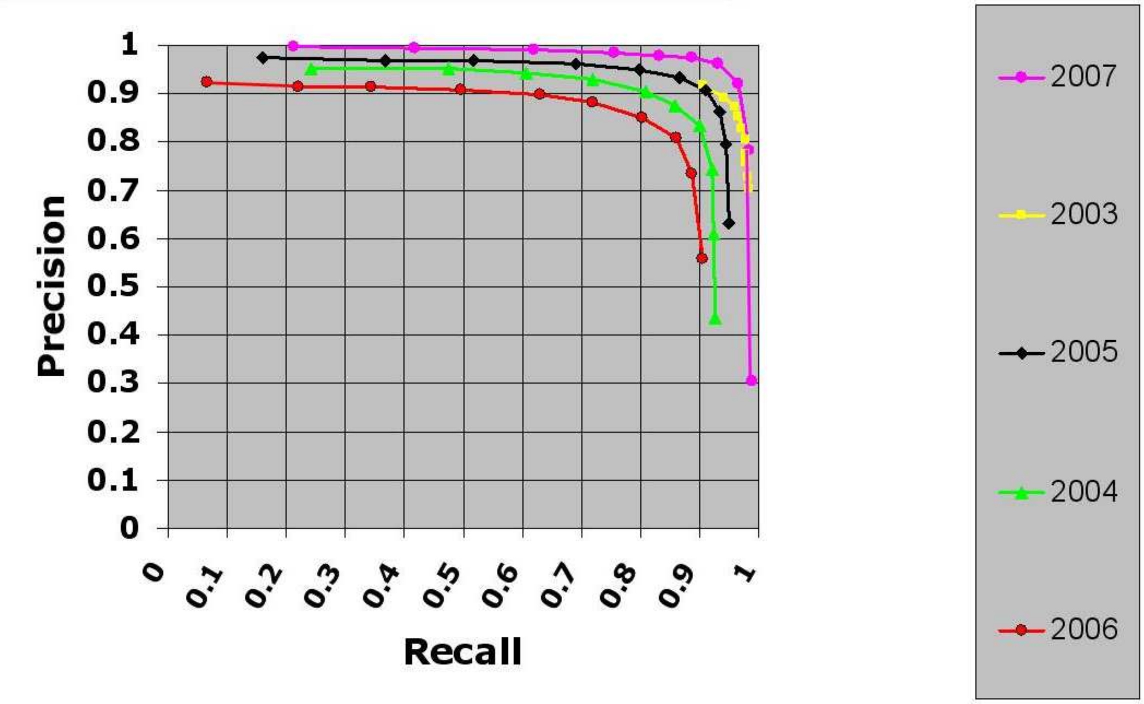


Fig. 4. Varying levels of difficulty of TRECVid gradual transitions, 2003-2006

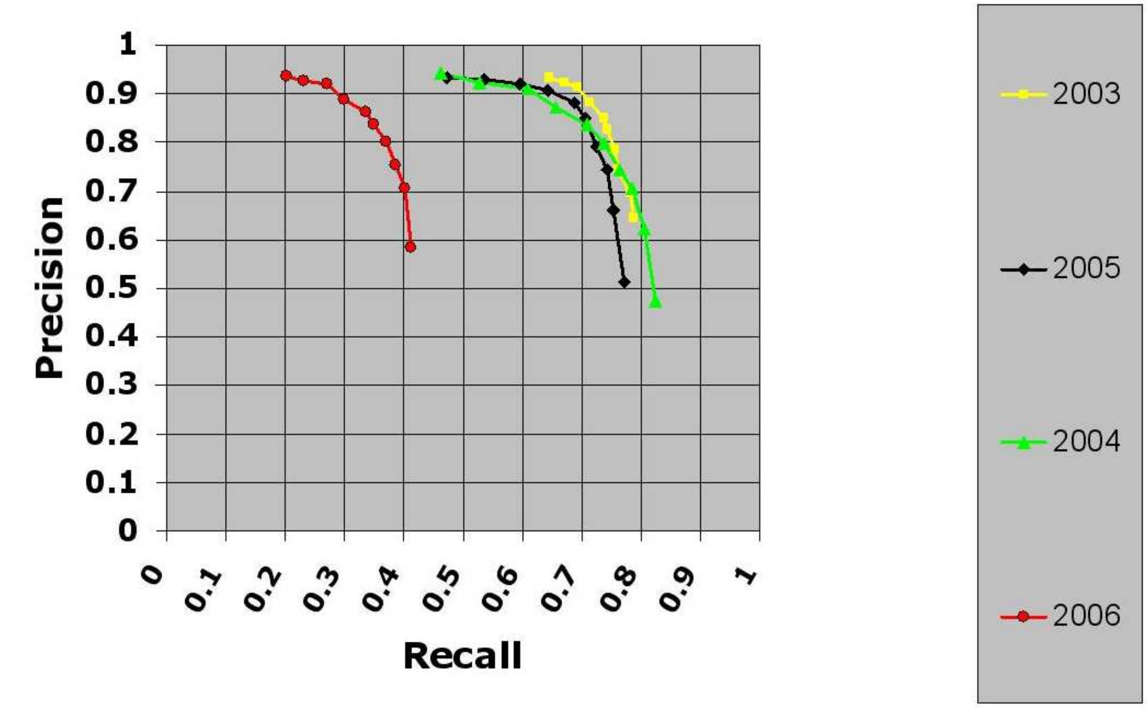


Table 6
F-measure for best-performing run from the CLIPS-IMAG group

Year	Cuts	Gradual Transitions
2003	0.917	0.774
2004	0.866	0.767
2005	0.911	0.763
2006	0.780	0.512
2007	0.947	-

show the performance figures for the best run from the top 10 groups in 2005 for cuts and for gradual transitions respectively, ranked by F score. What is immediately clear from these results is that there is very little difference in performance among those top groups, and in section 5.2 we will examine whether the differences among these are significant or not. The performance figures for hard cuts are particularly good, with 7 of the 10 having a F score of greater than 0.9. The performance of cut detection is significantly better than the performance of detection of gradual transitions, and this is as expected.

It is interesting to correlate the entries in Tables 7 and 8 with Table 3 which lists the approaches taken by participating groups, and we see that these top 10 in cuts and in gradual transitions cover a wide variety of approaches to SBD (e.g. machine learning, colour histograms, flash detection, working in the compressed domain, adaptive thresholding techniques, luminance values, edge histograms, and motion compensation). It is also interesting to see that 7 of the top 10 performing groups in cut detetcion also appear in the top 10 for detection of gradual transitions, and vice-versa, indicating that what works well for one, also works well for the other.

When we look at execution speed and correlate this with performance we would expect that there is a tradeoff in F score as against execution time, which is listed in Table 4 but in fact groups like KDDI R&D and the University of Marburg perform only slightly worse than NICTA, for example, in terms of F score, but are *much* faster in execution speed. This indicates that there is in fact no tradeoff between performance and speed, across different approaches taken.

5.2 Comparison using randomization testing

The tables of results listing precision, recall and their combination in the F-measure for hard cuts (Table 7) and for gradual transitions (Table 8) indicate a ranking of the top 10 systems, but not whether the differences among these systems are significant, i.e. likely to be due to chance rather than to real

Table 7
Top 10 systems for cuts in 2005

Recall	Precis.	F-score	Group
0.936	0.949	0.942	KDDI
0.930	0.941	0.936	TsinghuaU
0.947	0.914	0.930	NICTA
0.917	0.929	0.923	RMIT
0.924	0.900	0.912	U. Marburg
0.936	0.890	0.912	IBM
0.910	0.908	0.909	IMAG
0.951	0.842	0.893	U. Modena
0.951	0.842	0.893	CityUHK
0.919	0.828	0.871	TU Delft

Table 8
Top 10 systems for graduals in 2005

Recall	Precis.	F-score	Group
0.788	0.791	0.790	Tsinghua U.
0.773	0.781	0.777	NICTA
0.838	0.722	0.776	IBM
0.688	0.881	0.773	IMAG
0.741	0.790	0.765	FXPal
0.722	0.691	0.706	U. Marburg
0.686	0.727	0.706	Imperial
0.729	0.671	0.699	U. Modena
0.758	0.635	0.691	U. SaoPaolo
0.732	0.645	0.686	RMIT U

system differences. To address this question, a partial randomization test [18] was performed on the 2005 file-by-file results using the F-measure, separately for cuts and for gradual transitions.

For each pair of systems in these top 10, the randomization test generates a distribution of differences between the means of the file-by-file F scores

Table 9
2005 randomization test results for top 10 systems' cuts

KDDI	1	=	-	-	-	-	>>	>	>>	>>	>>
Tsinghua	2	-	=	>	-	>	>>	>	>>	>>	>>
NICTA	3	-	-	=	-	-	>	-	>>	>>	>>
RMIT	4	-	-	-	=	-	>	-	>>	>>	>>
Marburg	5	-	-	-	-	=	-	-	>>	>>	>>
IBM	6	-	-	-	-	-	=	-	>	-	>
CLIPS	7	-	-	-	-	-	-	=	>>	>	>
Florida/Modena	8	-	-	-	-	-	-	-	=	-	-
City Hong Kong	9	-	-	-	-	-	-	-	-	=	-
TU Delft	10	-	-	-	-	-	-	-	-	-	=
		1	2	3	4	5	6	7	8	9	10

Table 10
2005 randomization test results for top 10 systems' gradual transitions

Tsinghua	1	=	>>	>>	>>	>>	>>	>>	>>	>>	>>
NICTA	2	-	=	-	-	-	-	-	>>	>	>
IBM	3	-	-	=	-	-	>	>>	>>	>>	>>
CLIPS	4	-	-	-	=	-	-	>>	>>	>	>
FXPal	5	-	-	-	-	=	>	>>	>>	>>	>>
Marburg	6	-	-	-	-	-	=	-	-	-	-
Imperial	7	-	-	-	-	-	-	=	-	-	-
Florida/Modena	8	-	-	-	-	-	-	-	=	-	-
Sao Paulo	9	-	-	-	-	-	-	-	-	=	-
RMIT	10	-	-	-	-	-	-	-	-	-	=
		1	2	3	4	5	6	7	8	9	10

under the null hypothesis that the difference is due to chance. Then it counts how many such differences are equal to or more extreme than the observed difference. This count divided by the total number of generated differences, is taken as the probability that the observed difference in means is due to chance.

A program generates a distribution under the null hypothesis by calculating the difference for each file's pair of observed F scores and then iterating some tens of thousands of times randomly choosing for each pair of scores (for a file) whether to multiply the difference by -1 or 1 (i.e. to reverse the score-to-run assignment or not) before summing, calculating the mean, and finding the difference in the means. If there is no difference between the two systems then the score for a file from one system could just as well have been from the other system.

Each of the tables 9 and 10 shows the results of such a partial randomization test. The symbol ">" indicates the row system (the best performing run from each of the top 10 groups, using the run code in the tables rather than the group name) performed significantly better than the column system with a probability of less than 0.05 that the difference was due to chance ($p < .05$). The symbol ">>" indicates the row system performed significantly better than the column system with a probability of less than 0.01 that the difference was due to chance ($p < .01$). The symbol "-" indicates the row system did not perform better than the column system according to the tests.

If there had been a significant difference between all pairs of best runs then the tables would have consisted of >> symbols but this is not the case. The tables do indicate that while each system may not be better than its immediate successor in the ranking, it is significantly better than those a few places further down. This implies that the rank ordering among groups is a partial ranking rather than an absolute one.

6 Conclusion

It is challenging at best to try to draw concrete conclusions from such an enormous amount of experimentation from 57 research groups over 7 years and representing approximately 1,000 experimental runs, all exploring techniques for shot boundary detection from video. In this paper we have summarised the activities over these 7 years and have identified the evolution of the task in that period, focusing on one year in particular and giving a far more detailed analysis than the annual overview papers. What we have seen is a large variety of approaches and despite the fact that the data has generally gotten more difficult over the years, we observe excellent performance on cuts and gradual transitions, and although some might argue that 79% average for precision and recall is not excellent we disagree and think this is good performance. We also observe that good effectiveness is achievable at significantly less than realtime for many groups.

However, despite the continued introduction of novel approaches each year, which is documented in the annual TRECVid overview papers [19], [20]. [21], [22], [23] and in the workshop papers from individual participants, these novel approaches do not lead to improvements in effectiveness with the best systems achieving very good performance even on most gradual transitions. This has been true for the last few years of the TRECVid SBD task, and indeed as a community we decided to discontinue this task from TRECVid 2008 onwards. Also, the nature of an annual benchmarking activity like TRECVid is that it seeks to favour inclusiveness of participation with more participants each year and with different datasets from year-to-year, rather than always operating on the same fixed collection and being able to explore the SBD the task in a more scientifically rigourous and longitudinal way. A consequence of this is that it is not easy for us to identify exactly what components of the SBD process contribute most to solving the SBD problem but this is a small downside when we consider how the activity as a whole has helped to advance the state of the art in shot boundary detection.

Acknowledgment

The authors would like to thank the sponsors, John Lasko for doing the groundtruth creation, and the participating groups over the last 7 years. AS and AD are sponsored by Science Foundation Ireland under grant numbers 03/IN.3/I361 and 07/CE/I1147.

References

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

[2] R. Joyce, B. Liu, Temporal segmentation of video using frame and histogram space, IEEE Transactions on Multimedia 8 (1) (2006) 130–140.

[3] H. Zhang, A. Kankanhalli, S. Smoliar, Automatic partitioning of full-motion video, Multimedia Systems 1 (1993) 10–28.

[4] H. Lu, Y. Tan, An effective post-refinement method for shot boundary detection, IEEE Transactions on Circuits and Systems for Video Technology 15 (11) (2005) 1407–1421.

[5] W. Heng, K. Ngan, Shot boundary refinement for long transition in digital video sequence, IEEE Transactions on Multimedia 4 (4) (2002) 434–445.

[6] Y. Tan, J. Nagamani, H. Lu, Modified Kolmogorov-Smirnov metric for shot boundary detection, Electronics Letters 39 (18) (2003) 1313–1315.

[7] J. Nesvadba, F. Ernst, J. Perhavc, J. Benois-Pineau, L. Primaux, Comparison of shot boundary detectors, in: Int. Conf. for Multimedia and Expo, Amsterdam, The Netherlands, June, 2005, pp. 6–8.

[8] A. F. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and TRECVid, in: MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, ACM Press, New York, NY, USA, 2006, pp. 321–330.

[9] A. Lee, VirtualDub home page, http://www.virtualdub.org/index.html.

[10] R. Ruiloba, P. Joly, S. Marchand-Maillet, G. Quenot, Towards a Standard Protocol for the Evaluation of Video-to-Shots Segmentation Algorithms, European Workshop on Content Based Multimedia Indexing. Toulouse, France: URL: clips. image. fr/mrim/georges. quenot/articles/cbmi99b. ps.

[11] Y. Ma, J. Sheng, Y. Chen, H.-J. Zhang, MSR-Asia at TREC-10 Video Track: Shot Boundary Detection Task, Proceedings of TREC-10.

[12] A. R. Doherty, A. F. Smeaton, Automatically segmenting lifelog data into events, in: WIAMIS: 9th International Workshop on Image Analysis for Multimedia Interactive Services, IEEE Computer Society, Washington, DC, USA, 2008, pp. 20–23.

[13] K. McDonald, A. F. Smeaton, A comparison of score, rank and probability-based fusion methods for video shot retrieval, in: CIVR 2005 - International Conference on Image and Video Retrieval, W-K Leow et al. (Eds.), LNCS 3568, pp61-70, Springer, 2005, pp. 61–70.

[14] C. Cai, K. M. Lam, Z. Tan, TRECVID2005 Experiments in the Hong Kong Polytechnic University: Shot Boundary Detection Based on a Multi-Step Comparison Scheme, in: TRECVID 2005 Workshop Notebook Papers, National Institute of Standards and Technology, MD, USA, 2005.

[15] A. Lewis, G. Knowles, Video compression using 3D wavelet transforms, Electronics Letters 26 (6) (1990) 396–398.

[16] A. Amir, The IBM Shot Boundary Detection System at TRECVID 2003, in: TRECVID 2005 Workshop Notebook Papers, National Institute of Standards and Technology, MD, USA, 2003.

[17] G. Quenot, D. Moraru, L. Besacier, CLIPS at TRECVid: Shot Boundary Detection and Feature Detection, TRECVID 2003 Workshop Notebook Papers, Gaithersburg, MD, USA (2003) 18–21.

[18] B. Manly, Randomization, Bootstrap and Monte Carlo Methods in Biology (2nd Ed.), Chapman and Hall, London, UK, 1997.

[19] A. Smeaton, W. Kraaij, P. Over, TRECVID 2003-an overview, TRECVID 2003 Workshop Notebook Papers, Gaithersburg, MD, USA.

[20] W. Kraaij, A. Smeaton, P. Over, TRECVID 2004-An Overview, TRECVID 2004 Workshop Notebook Papers, Gaithersburg, MD, USA.

[21] P. Over, T. Ianeva, W. Kraaij, A. Smeaton, TRECVID 2005-An Overview, TRECVID 2005 Workshop Notebook Papers, Gaithersburg, MD, USA.

[22] P. Over, T. Ianeva, W. Kraaij, A. Smeaton, TRECVID 2006-An Overview, TRECVID 2006 Workshop Notebook Papers, Gaithersburg, MD, USA.

[23] P. Over, G. Awad, W. Kraaij, A. Smeaton, TRECVID 2007-Overview, TRECVID 2007 Workshop Notebook Papers, Gaithersburg, MD, USA.