

Adaptive Acceleration and Shot Stacking for Video Rushes Summarization

Marcin Detyniecki
Université Pierre et Marie Curie Paris6
CNRS UMR 7606, LIP6,
104 av. du Président Kennedy
Paris, F-75016, France
Marcin.Detyniecki@lip6.fr

Christophe Marsala
Université Pierre et Marie Curie Paris6
CNRS UMR 7606, LIP6,
104 av. du Président Kennedy
Paris, F-75016, France
Christophe.Marsala@lip6.fr

ABSTRACT

As the amount of recorded video data continually increases, a lot of teams around the world work to propose original methods on automatic video summarization. A particular kind of video is a rush: a rough draft of a movie or a documentary. In this paper, an approach is proposed to summarize the BBC rushes within the scope of the 2008 TREC Video Retrieval Evaluation task. We propose to summarize the videos by applying two successive steps, each dealing with a particular reduction. First, the stacking step focuses on the reduction of the macro redundancy with respect to similar takes of the same scene (rushes). The second step focuses on the time redundancy of the video flow and can be interpreted as an adaptive acceleration. This approach provides good inclusion rates.

Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—*Video*; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—*Evaluation/Methodology*

General Terms

Algorithms, Experimentation, Performance.

Keywords

Video summarization, TRECVID 2008, adaptive acceleration, shot stacking.

1. INTRODUCTION

Nowadays, there is increasing interest in automatic video summarization. The amount of recorded video data is continually increasing leading to a growing need to find a solution of how to handle it. Some years ago, a specific TREC Video Retrieval Evaluation task was introduced focusing on

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

TVS'08, October 31, 2008, Vancouver, British Columbia, Canada.
Copyright 2008 ACM 978-1-60558-309-9/08/10 ...\$5.00.

the summarization of video, with one singularity: the type of data. The BBC Archive provided unedited materials in order to enable the community to compare and to study several kinds of approaches of summarization.

Video rushes are rough drafts of a movie or a documentary. They are composed of all the filming that was recorded and from which the final document is made. It is usually a highly redundant dataset with a lot of duplicated or slightly different sequences. In the particular case of the TRECVID BBC rushes, each video is around 30 minutes long, and it can contain several unmarked rushes of different scenes. The aim of the challenge is to summarize a video from the TRECVID BBC rushes such that the summary will have a duration which is at most two percent (2%) of the original video [3, 4, 5]. In 2007, several groups participated to the challenge and proposed original solutions to solve this problem [1]. However, the problem stays open and most of the approaches deserve some enhancements. Thus, the challenge was proposed again in 2008.

In this paper, we present our approach to summarize rushes, which by nature differs from other forms of video summarization. In Section 2, our approach for summarizing BBC rushes is presented. We focus on the specificity of our approach: a two step reduction; first stacking of shots and then an adaptive acceleration. In Section 3, the results obtained in the TRECVID 2008 challenge are presented and discussed.

2. SUMMARIZING BBC RUSHES

A basic way to produce a video that fulfills the two-percent duration constraint is to randomly delete frames of the original video in order to keep only 2% of them (*e.g.* keeping one frame every 50 consecutive frames). However, this approach is not convenient if the aim is (1) to keep, from the original video, as much information as possible, (2) to delete junk shots and useless frames, and (3) to avoid redundancy in the summary. Thus, we propose to take into account the *information* provided by each frame.

Similar to the approach we presented in the TRECVID 2007 BBC rushes summarization challenge [2], this year we focus on keeping as much information as possible, but respecting the two-percent duration limit.

Our method is based on four steps. First, a shot-boundary detection transforms the video into a set of shots. Secondly, a comparison of successive shots is done, enabling us to *stack* similar shots (*i.e.* shots that mainly contain the same kind of information). From each stack, a particular shot is *se-*

lected and will be kept in the summary of the video. The other ones will be forgotten since they are considered to contain the same information as the selected one. Third, an adaptive acceleration deletes, within elected shots, as much as possible redundant temporal information. The final summary is the results of a video reconstruction that targets the two percent duration constraint of the challenge. In the following, we explain in detail each step of this method.

2.1 Shot detection

Although it seems possible to measure a degree of informativeness directly from the compressed data flow, for the sake of simplicity, we choose to work with images obtained by sampling uniformly the video. In a second step these images are visually described.

The original MPEG video file is sampled using *ffmpeg*¹. A constant frames per second (fps) rate was chosen to sample the video. The idea is to reduce the number of frames to be treated and thus processing time of the forthcoming steps.

From each frame we extract a set of low-level descriptors. For a global visual comparison we calculate the HSV color histogram of the frame. In addition, each frame is segmented into distinct regions. For each region we compute its HSV color histogram.

Two frames are compared by means of the Euclidean distance² of their histograms: for a given region, for a set of regions, or for the whole image.

Although some preliminary tests show that the combination of global and region based descriptors improve the results [2], the choice of the number of regions and the choice of the number of bins for the histogram require further optimization.

To find shots in the video, the similarity between successive frames is used. Since it is not the heart of our approach (and clearly deserves further optimization), a basic shot detection algorithm was implemented. The shot boundaries are valued by the variations in terms of similarity on a short list of successive frames.

As a rule of thumb, a shot boundary is considered when the average of the similarities between several pairs of consecutive frames increases dramatically. The use of the average over a number of successive pairs, makes the approach robust to several kinds of transitions (e.g. fade in, fade out, strong cut, ...).

Since the use of a similarity based only on the global histogram leads to poor transition detection (because it is too sensitive to changes in a frame), the similarity between frames is valued by considering the different region-similarities.

The detected boundaries enable us to define a set of shots that represents the whole video. As a first pruning step, very short shots (duration of less than around 2 seconds) are considered intrinsically non-informative and removed from the set of shots.

2.2 Stacking shots

In our approach, rushes are defined as similar pieces of video. Rushes are typically re-takes of the same scene. Thus in order to summarize, grouping similar rushes is a seminal

¹All the processing of the MPEG files are done with the software *ffmpeg* available at <http://ffmpeg.mplayerhq.hu/>

²There are a lot of approaches to measure the similarity between frames, which could be used instead of the Euclidean distance.

step. The most common way [1] to do this is to apply a clustering algorithm to the whole set of shots of the video, eventually leading to high processing times. Moreover, the clustering algorithm should be stated carefully by determining its parameters and, in particular, the number of clusters (*i.e.* number of different rushes) that should be found in the video.

Since 2007, we propose another approach that takes into account the specificity of the BBC rushes. As a general rule, rushes in the BBC videos are more or less successive. In fact, BBC rushes are generally recorded one after the other. Thus, only a small number of successive shots needs to be compared. Furthermore, it is sufficient to look locally at some successive shots, in order to determine if they pertain to the same rush. This specific solution is faster than clustering-based ones, and provides good results, as shown in our experiments.

2.2.1 Characterization of shots

At this point, for each shot, a subset of characteristic frames is extracted to represent the shot for further comparisons. A given number of characteristic frames have been chosen both, at the beginning and at the end of the shot. A characteristic frame is a frame that strongly differs from its preceding ones. To find such a frame, a frame-by-frame comparison is done. Here we use the maximum of the Euclidean distance between the region histograms.

Depending on where they have been detected (either at the beginning or at the end of the shot), these characteristic frames are called either *beginning frames* or *ending frames*.

2.2.2 Comparing and grouping the shots

The aim of our method is to highlight similarities between shots in order to regroup them as the same rush. The similarity between two shots is valued by aggregating the frame similarities of the *beginning frames* set and *ending frames* set. Since here, the similarity of the frames can be valued roughly, we consider the global histograms.

Two shots are considered similar if one of the following is true (*i.e.* above a certain threshold): (1) the beginnings are similar, (2) the endings are similar, and (3) the beginning of one shot is similar to the ending of another one. The underlying intuition is based on the observation that rushes are often, either stopped before the scene is completely played to the end, or played from a particular point to the end.

As a result, similar shots will be grouped to constitute a kind of stack of shots. At the end of this stacking process, when all the shots have been processed and compared to previous ones, a set of stacks of shots is provided.

From each stack, the longest shot (in duration) is selected as being the most informative one. And it will be the only shot of the stack displayed in the final summary. It can be noted here that, due to the stacking look-ahead process, the temporal order of two representative shots from two successive stacks is not necessarily conserved.

2.3 Adaptive Acceleration

The idea for an adaptive acceleration, in a video summarization context, is to erase successive non-informative frames. The remaining frames are then shown at a constant rate. As a result, when not much is happening on the screen less time is used to present it.

The adaptive acceleration used this year is similar to the

one used in [2]. Frames, taken successively, are compared with a base frame (at the beginning of the process, the base frame corresponds to the first frame). If the forthcoming frame is similar to the base one (*i.e.* they just slightly differ) then, it can be considered that, the forthcoming frame does not bring about a lot of additional information, and can be forgotten and rejected when displaying the summary. This process is repeated until a frame is dissimilar enough to the base one. In that case, it is kept and becomes the base one.

During the acceleration process, the informative distance is valued by means of the visual differences that appear in regions of the frames. First of all, the visual difference of two corresponding regions of the two frames is valued as the Euclidian distance of the respective HSV histograms. Afterwards, the visual differences between all the regions that have been chosen in the frames are combined by means of the maximum in order to value the informative distance between the two frames. Two frames are considered similar if their informative distance is below a given threshold³.

In [2], a maximum look-ahead value was set to limit the deletion of forthcoming frames: after a given number of deletion of forthcoming frames, the next frame was kept even if it was similar to the base frame. This year, by fixing no limit in the number of deleted frames, a so-called *infinite* acceleration was used.

2.4 Reconstructing the video

Once the subset of informative frames (of the representative shots) was extracted, the video summary is reconstructed using ffmpeg. In order to obtain a video from a set of frames only the number of frames to be played per second is needed. In our experiment, the frame rate was calculated to meet the challenge’s 2% compression constraint on the duration of the original video.

3. RESULTS AND DISCUSSION

3.1 Global results of the method

Table 1 presents the global results obtained by our approach, compared to results obtained by all other teams (32 teams that submitted, in total, 43 runs).

Our approach is characterized by a high (with respect to others) fraction of inclusions found in the summary (IN in Table 1). In other words, the resulting summaries tend to keep a high percentage (67%) of the information contained in the initial video. The main reason is that, during our summarization process, every selection was designed with the purpose of keeping as much information as possible. Staggeringly this year no method provided a better inclusion rate than the baseline (IN=0.83).

Moreover, we observed over all our runs that, in average, there is a 0.13 variation in the inclusion value, when comparing the different assessors. Which means that there is a disparity in their appreciation about the fact that the summary kept or not some information.

Our summaries meet by construction (cf. Section 2.4) the 2% duration constraint. Therefore, the difference between target and actual summary size (XD=1.26s) should equal zero. By looking at the XD values, we notice that most of the participants had difficulties in meeting the constraint

³Thresholds were experimentally estimated using the DEVEL data set.

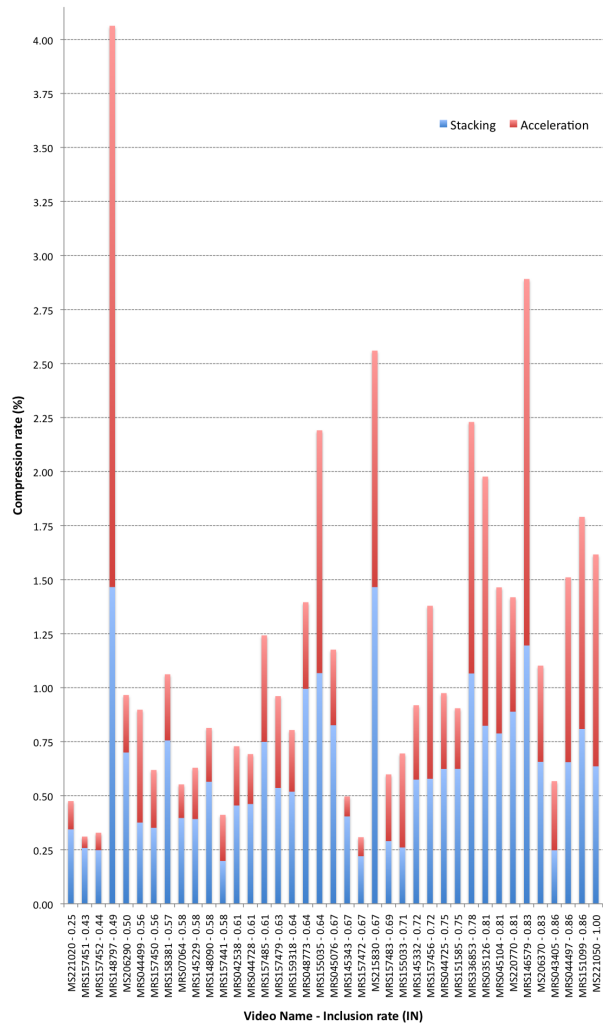


Figure 1: Compression rate for each video of the corpus, with relative contribution of the stacking and acceleration steps. The video summaries are ordered by increasing inclusion rate.

and submitted longer summaries. Moreover, most of our summaries (35 of 39) met the 2% constraint as result of the stacking and acceleration steps (see Figure 1). Thus, the reconstruction slowed down the playback tempo. The average compression rate, before the reconstruction, is 1.17%.

The total time spent to judge our summaries (TT=50.42s) is long compared to others. There seems to be a relation between this time and the inclusion rate. In fact, if we consider the total video playtime judging the inclusions (VT=31.16s) and the duration of the summary (DU=30.45s) we note that most of the judging time (TT) was spent on pause. We presume that the time was used to mark that some inclusion was found. Moreover, this values point out that the assessors did not have any particular difficulties judging the summary, or at least did not replay the video.

Our construction time is in average, for a summary, 3660 seconds - below the all team’s average of 4879 seconds. When compared to the initial video duration, it takes our system about 2.4 times the real time. Most of this time is con-

Table 1: Results for the BBC rushes task: UPMC-LIP6 and all teams

		TT (secs)	VT (secs)	DU (secs)	XD (secs)	IN (0 - 1)	JU (1 - 5)	RE (1 - 5)	TE (1 - 5)	Build. time (secs)
	UPMC-LIP6	50.42	31.16	30.45	1.26	0.67	2.52	2.6	1.85	3660
Mean	Avg. (all)	41.2	29.36	27.11	4.6	0.44	3.16	3.27	2.73	4879
	Max (all)	59.59	53.25	47.81	18.15	0.83	3.64	3.99	3.38	41556
	Min (all)	22.56	19.56	13.56	-16.1	0.07	2.52	2.02	1.44	17

sumed by the extraction, from the video, of the key-frames and their associated color histograms.

Since the stacking and the adaptive acceleration do not perform any complicated calculations, and since the data set or the video are never considered as whole, our approach scales linearly with the number and with the length of the videos to be summarized.

The assessors considered that our approach provided a lot of junk (JU=2.52), which is understandable since we did not have any filters for junk shots, as for instance claps views.

3.2 Stacking versus Adaptive Acceleration

As we observe on Figure 2 that the contribution of the stacking and the acceleration steps is equilibrated. In fact, although the compression strength of the stacking (in average 43% of the frames are kept) is weaker than the one of the acceleration (in average 2.7% are kept), the global contribution is similar since the acceleration is applied after the stacking.

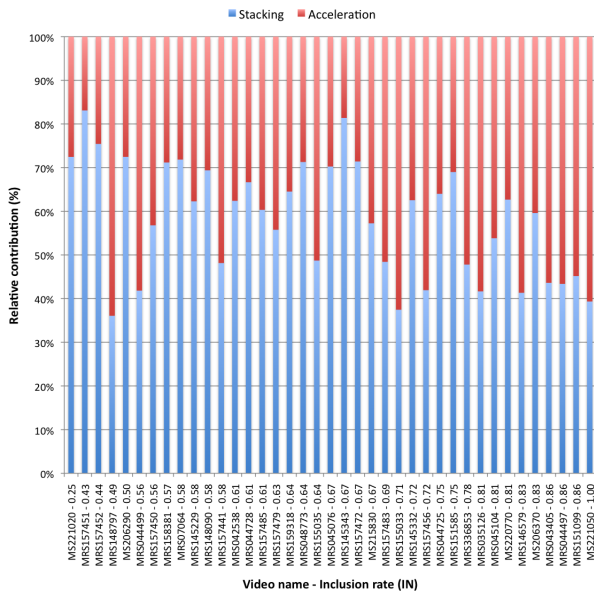


Figure 2: Stacking vs adaptive acceleration. The video summaries are ordered by increasing inclusion rate.

The stacking step tackles the summarization by acting on the redundancy of similar rushes. The relatively low score (RE=2.6), on the question “was there a lot of duplicate video?”, suggests that in our case there is a negative perception of the redundancy. However, as shown on Figure 3, there is no correlation between this score and the actual

stacking ratio (i.e. how much was eliminated on this step). Thus, we believe that this perception is due to other factors, as the inclusion rate and the playing tempo. This question needs further investigation.

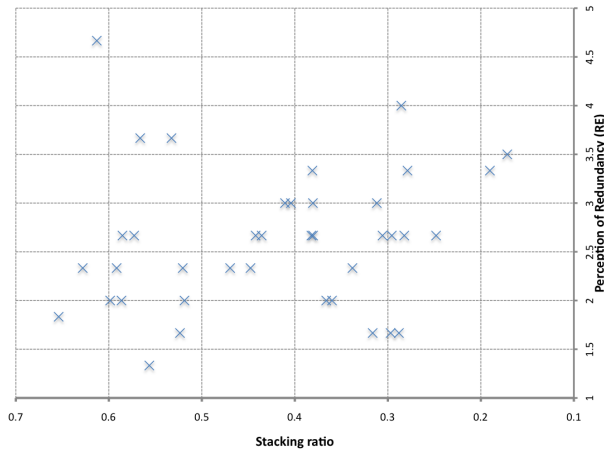


Figure 3: Stacking ratio and Perception of redundancy (RE) for the 39 UPMC LIP6 summaries.

If we focus on the relationship between stacking and inclusion ratio we observe, on Figure 4, a weak correlation: the more we stack the more we loose in inclusion. Nevertheless, a linear regression indicates a small decrease ratio: 0.05 inclusion ratio variation for each 10 percent point change of the stacking ratio. In addition, we observe a low determination coefficient (0.26).

Even though the acceleration ratio is much stronger than the stacking ratio, there is no correlation between the actual acceleration ratio and the inclusion ratio. Which suggests that the acceleration preserves quite well the information.

Although our approach has an adaptive acceleration, we can measure an average acceleration rate. For our summaries this rate ranges from 8 times to 32 times the real time. In average, the assessors gave a rather negative score (TE=1.85) when asked the question “summary has a pleasant tempo - rhythm?”. This can be interpreted as that an acceleration rate of more than 8 times is definitively unpleasant, but as seen on Figure 5, the perception of the tempo (TE) is unrelated to the actual (average) acceleration rate. Which suggests that there is no correlation between the real tempo and the perceived one. Furthermore, we believe that the perception of the tempo (TE) is closely related to the inclusion ratio (IN). Figure 6 is evidence for a strong correlation when considering not only our runs, but all the groups.

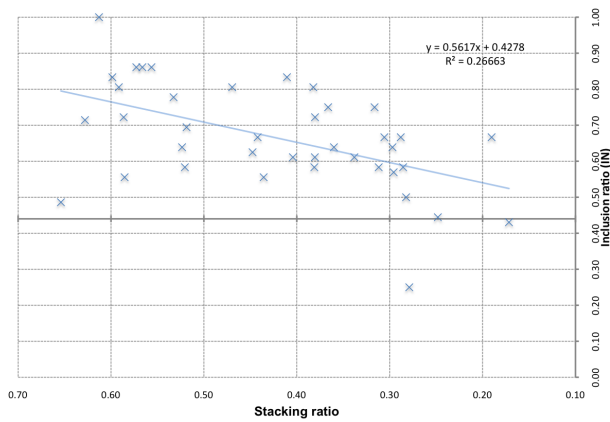


Figure 4: Variation of the inclusion ratio (IN) in relation with stacking.

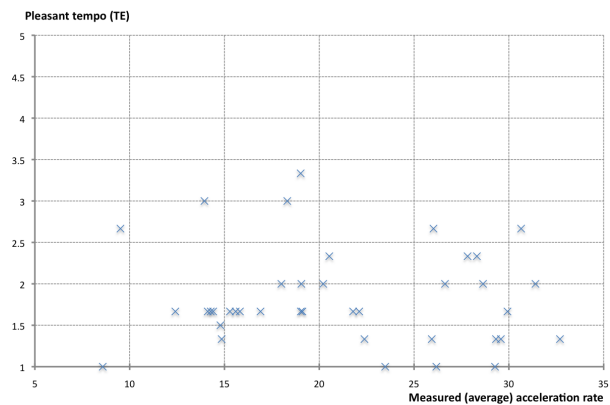


Figure 5: Relation between measured average acceleration rate and perception of the tempo (TE) based on the 39 UPMC LIP6 summaries.

A linear regression provides a relatively high determination coefficient (0.55) and the variation ratio corresponds to a 0.25 TE drop for every 0.1 IN increase.

Based on the previous observations we believe that the interpretation of the TE and RE scores have to be considered with circumspection. Since identifying *separate* subjective criteria is a challenging task, we suggest adding the global appreciation question: How good the summaries are?

4. CONCLUSION

In this paper we presented the University Pierre and Marie Curie (UPMC LIP6) approach submitted to the TrecVid 2008 BBC rush video summary challenge. We proposed to summarize the videos by applying two successive steps, each dealing with a particular reduction. First, the stacking step focuses on the reduction of the macro redundancy with respect to similar takes of the same scene (rushes). The second step focuses on the time redundancy of the video flow and can be interpreted as an adaptive acceleration. This approach provides good inclusion rates (IN) compared to other state of the art systems and naturally respects the 2% reduction constraint.

The assessors judged severely our method, in terms of

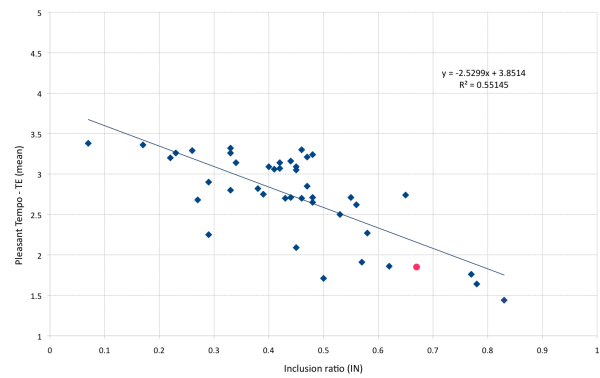


Figure 6: Relation between Perceived tempo (TE) and Inclusion ratio (IN) for all participating groups. The circle represents the UPMC LIP6 submission.

their perception of a pleasant tempo and their perception of redundancy. We presented here a study that shows that these two subjective scores are not correlated either to the measured acceleration rate, or to the compression ratio resulting of the stacking. Furthermore, there is evidence that the two scores are related to the inclusion rate.

5. REFERENCES

- [1] ACM. *TRECVID Workshop on Video Summarization (TVS'07 at ACM Multimedia'07)*, Augsburg, Germany, September 2007.
- [2] M. Detyniecki and C. Marsala. Video rushes summarization by adaptive acceleration and stacking of shots. In *Proceedings of the ACM international workshop on TRECVID video summarization*, pages 65–69, Augsburg, Germany, 2007.
- [3] P. Over, A. F. Smeaton, and G. Awad. The TRECVID 2008 BBC rushes summarization evaluation. In *TVS'08: Proceedings of the International Workshop on TRECVID Video Summarization*, pages 1–??, Vancouver, British Columbia, Canada, 2008. ACM, New York, NY, USA.
- [4] P. Over, A. F. Smeaton, and P. Kelly. The TRECVID 2007 BBC rushes summarization evaluation pilot. In *Proc. of the TRECVID Workshop on Video Summarization (TVS'07)*, pages 1–15. ACM Press, New York (NY), September 2007.
- [5] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVID. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.