IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, May 2014.

FAST SHOT SEGMENTATION COMBINING GLOBAL AND LOCAL VISUAL DESCRIPTORS

Evlampios Apostolidis, Vasileios Mezaris

Information Technologies Institute / Centre for Research and Technology Hellas, Thermi 57001, Greece
{apostolid, bmezaris}@iti.gr

ABSTRACT

This paper introduces an algorithm for fast temporal segmentation of videos into shots. The proposed method detects abrupt and gradual transitions, based on the visual similarity of neighboring frames of the video. The descriptive efficiency of both local (SURF) and global (HSV histograms) descriptors is exploited for assessing frame similarity, while GPU-based processing is used for accelerating the analysis. Specifically, abrupt transitions are initially detected between successive video frames where there is a sharp change in the visual content, which is expressed by a very low similarity score. Then, the calculated scores are further analysed for the identification of frame-sequences where a progressive change of the visual content takes place and, in this way gradual transitions are detected. Finally, a post-processing step is performed aiming to identify outliers due to object/camera movement and flash-lights. The experiments show that the proposed algorithm achieves high accuracy while being capable of faster-than-real-time analysis.

Index Terms— Shot Segmentation, SURF descriptors, HSV histograms, GPU processing

1. INTRODUCTION

Video shot segmentation aims to partition the video into groups of consecutive frames captured without interruption by a single camera. These elementary structural units, which are called shots, by definition demonstrate a certain degree of temporal and visual affinity, thus constituting a self-contained visual entity. Shot segmentation can be seen as the foundation of most high-level video analysis approaches, being a prerequisite for tasks such as video semantic analysis and fine-grained classification, indexing and retrieval.

For effective video segmentation to shots, two types of shot transitions need to be detected, abrupt and gradual ones. In the case of abrupt transitions, the last frame of a shot is followed by the first frame of the next shot, while in the case of a gradual transition there is a short intermediate temporal interval in which the visual content of the two consecutive shots is combined using one or more video editing effects such as fade in/out, wipe and dissolve. In addition to accurately detecting both of the aforementioned types of shot transitions, the required computation time is another critical property of a video segmentation algorithm, similarly to any algorithm that needs to be applicable to vast amounts of video content.

This work was supported by the European Commission under contracts FP7-287911 LinkedTV and FP7-318101 MediaMixer.

2. RELATED WORK

Early approaches to shot segmentation in uncompressed video were based on pair-wise pixel comparisons and/or the comparison of color histograms between video frames, calculated either for each entire frame or at a finer block level [1]. Other techniques exploited structural features of the frames, such as edges, and performed shot segmentation by estimating for instance the edge change ratio between frames [2]. Recent extensions of such methods include [3] and [4], where the video shots are detected based on the combination of edge information with color histograms and motion, respectively.

After the introduction of Support Vector Machines (SVM) [5] several approaches used them for shot segmentation, as a way of classifying video frames into "boundary" and "non-boundary". A method that compares normalized RGB histograms and uses a trained SVM classifier was introduced in [6], while another SVM-based approach that employs intensity pixel-wise comparisons, HSV histograms and edge histograms was proposed in [7]. Furthermore, visual features that are not typically used for video shot segmentation, such as the Color Coherence [8] and the Luminance Center of Gravity, were combined in a SMV-based approach for abrupt and gradual transition detection in [9].

Following the development of scale- and rotation-invariant local descriptors (e.g. SIFT [10] and SURF [11]), several algorithms that rely on them were also proposed. In [12] a technique based on the detection and tracking of objects in successive video frames was introduced. According to this, the SIFT descriptors extracted from the frames are clustered into a predefined number of bins, creating a codebook of visual words. Each frame is then represented by a histogram of words and the shot boundaries are determined based on the similarity of these histograms. In [13], a divide-and-rule scheme that combines SIFT descriptors and SVM classifiers for capturing the changing statistics of several kinds of transitions was proposed, while a different approach based on frame entropy and SURF descriptors was presented in [14].

Based on the needs for real-time multimedia analysis, we propose a shot segmentation method that ensures high levels of accuracy, while being able to complete the processing of a video in time corresponding to only a small fraction of the video's actual duration. The proposed approach relates to techniques such as [12] [13] [14] that rely on the use of local descriptors. However, the combination of SURF descriptors with HSV color histograms in a GPU-based computing framework that is proposed in this work has advantages compared to such previous approaches for shot segmentation, and has not been proposed in earlier works.

3. PROPOSED APPROACH

The proposed GPU-based approach identifies the shot boundaries by quantifying the change in the content of successive frames of the video, and comparing it against experimentally specified thresholds that indicate the existence of abrupt and gradual shot transitions. For this purpose, each frame is represented by an HSV color histogram and a set of SURF descriptors, allowing the algorithm to detect effectively the differences between a pair of frames, both in color distribution and at a more fine-grained structure level. The developed technique can be divided in three parts; the first one computes the similarity between the video frames and detects the abrupt transitions, the second one further processes the calculated similarity scores and detects the gradual transitions, and the third one filters-out possible erroneously detected shot boundaries due to object/camera movement and camera flash-lights. Each one of these parts is described in details in the following sub-sections. Finally, a simple fusion approach (i.e., taking the union of the detected abrupt and gradual transitions) is used for forming the output of the algorithm.

3.1. Pairwise frame similarity estimation and abrupt transition detection

For measuring the resemblance between a pair of frames (e.g. frames i and $i+1$) we use the formula:

$$\mathbf{F}(i) = \frac{d(H_i, H_{i+1}) + (1 - b(H_i, H_{i+1}))}{2} \frac{S_i + S_i'}{2} \quad (1)$$

In the above formula $\mathbf{F}(i)$ is the calculated similarity score, $d(H_i, H_{i+1})$ is the normalized correlation of the HSV histograms H_i and H_{i+1}, which have W bins (50 bins for hue and 60 for saturation), and $b(H_i, H_{i+1})$ denotes the Bhattacharryya factor for the same pair of histograms. $d(H_i, H_{i+1})$ and $b(H_i, H_{i+1})$ are computed as follows:

$$d(H_i, H_{i+1}) = \frac{\sum_{j=1}^{W} (H_i(j) - H_i')(H_{i+1}(j) - H_{i+1}')}{\sqrt{\sum_{j=1}^{W} (H_i(j) - H_i')^2 \sum_{j=1}^{W} (H_{i+1}(j) - H_{i+1}')^2}} \quad (2)$$

$$b(H_i, H_{i+1}) = \sqrt{1 - \frac{1}{\sqrt{H_i' H_{i+1}' W^2}} \sum_{j=1}^{W} \sqrt{H_i(j) H_{i+1}(j)}} \quad (3)$$

$$where\ H_i' = \frac{1}{W} \sum_{j=1}^{W} H_i(j) \quad (4)$$

The SURF Hessian factor, that affects the number of detected keypoints, was set to 400, while the matching of the calculated descriptors was performed in a brute force manner (i.e., each descriptor extracted from one frame was matched against all the descriptors extracted from the following frame), looking each time for the 2 best matches via k-Nearest Neighbor (k-NN) search. So, for each detected keypoint in frame i we search for the best matches in frame $i+1$ that correspond to the two nearest neighbors N_1 and N_2. Erroneous matches are then filtered-out based on the following rule: we keep a keypoint in frame i and its corresponding best match in frame $i+1$ if $\|N_1\| \ / \ \|N_2\| \leq 0.8$, where $\|\cdot\|$ is the Manhattan distance between the corresponding nearest neighbor and the keypoint. Then, the factors S_i and S_i' of Eq. (1) are computed as the ratio of the matched keypoints M_i to the number of detected keypoints in each of the compared video frames, K_i and K_{i+1}, respectively.

$$S_i = M_i / K_i \ \ , \ \ S_i' = M_i / K_{i+1} \quad (5)$$

The calculation and comparison of HSV histograms and the extraction and matching of SURF descriptors between pairs of frames was realized using version 2.4.7 of the OpenCV library. Aiming to speed-up the analysis we accelerated the parts related to local feature extraction and matching by utilizing the available GPU-based implementations.

Finally, based on the resulting similarity scores (Eq. (1)), an abrupt transition is declared in between every pair of frames with a similarity score lower than a threshold $T_a = 0.2$.

3.2. Detection of potential gradual transitions

For the detection of gradual transitions, we further analyse the computed similarity scores trying to identify patterns that correspond to progressive changes of the visual content over sequences of frames. Denoting $\mathbf{F}$ the curve (see Fig. 1) formed by the similarity scores computed in the first part of processing (Eq. (1)) for all frames, then the potential shot boundaries due to gradual transition are detected as described in Alg. 1.

The output of this algorithm is a vector $\mathbf{C}$ of scores that represent the visual dissimilarity between each frame that corresponds to a local minimum of the moving average curve $\mathbf{G}$ (Alg. 1) and two frames that correspond to local maxima of the same curve and surround the former local minimum. Video frames where the computed dissimilarity is higher than a threshold $T_b = 0.25$ are declared shot boundaries.

Algorithm 1 Detection of potential gradual transitions

```
Notation: T is the number of frames of the video, $\mathbf{F}(i)$, $i = 1...(T-1)$ are the similarity scores, $\mathbf{G}(k)$, $k = 1...(T-2V-1)$ is the moving average curve, $V = 5$ is the temporal window for computing $\mathbf{G}(k)$, $\mathbf{E}(k)$ is the first order derivative of $\mathbf{G}(k)$, vectors $\mathbf{G_{min}}$ and $\mathbf{G_{max}}$ store the local minima and maxima of $\mathbf{G}(k)$, $\mathbf{D}(k)$ is the dissimilarity vector, $\mathbf{C}(k)$ is the clustered dissimilarity vector, $R$ is the video's frame-rate and $Y = 0.15$.
Input: The similarity scores $\mathbf{F}(i)$.
Output: The clustered dissimilarity vector $\mathbf{C}(k)$.
1: Load $\mathbf{F}(i)$ (top graph in Fig. 1) and compute the moving average vector $\mathbf{G}(k)$ (middle graph in Fig. 1) as:
   for $k = 1 \to (T-2V-1)$
       $\mathbf{G}(k) = \sum_{m=-V}^{V} \frac{\mathbf{F}(k+V+m)}{2V+1}$
2: Calculate $\mathbf{E}(k)$ as the first order derivative of $\mathbf{G}(k)$ and store the local minima and maxima of $\mathbf{G}(k)$ in vectors $\mathbf{G_{min}}$ and $\mathbf{G_{max}}$, respectively
3: Calculate $\mathbf{D}(k)$ (bottom graph in Fig. 1) as:
   for $k = 1 \to (T-2V-1)$
       if frame $k$ is the $p-th$ element of $\mathbf{G_{min}}$ then $\mathbf{D}(k) = |\mathbf{F}(\mathbf{G_{min}}(p)) - \mathbf{F}(\mathbf{G_{max}}(p-1))| + |\mathbf{F}(\mathbf{G_{min}}(p)) - \mathbf{F}(\mathbf{G_{max}}(p)|$, else $\mathbf{D}(k) = 0$
4: Calculate $\mathbf{C}(k)$ (bottom graph in Fig. 1), as:
   for $k = 1 \to (T-2V-1)$
       if $\mathbf{D}(k) > Y$ then $\mathbf{C}(k) = \sum_{l=-R/2}^{R/2} \mathbf{D}(k+l)$ (providing that $\mathbf{D}(k+l)$ exists), else $\mathbf{C}(k) = 0$
```

3.3. Identifying outliers due to object / camera movement

As depicted in the middle and bottom graphs of Fig. 1, some of the detected shot boundaries might correspond to sequences of frames exhibiting object or camera movement (and a similar pattern was observed in the case of camera flashes). Aiming to identify these outliers, we evaluate the similarity of each detected boundary against the pair of frames that correspond to its previous and following local maxima from vector $\mathbf{G_{max}}$, by re-applying the formula of Eq. (1). If the calculated score is higher than a threshold $T_c = 0.10$, we consider that the compared frames belong to the same shot and we remove the detected shot boundary, while otherwise we retain it.

4. EXPERIMENTS AND RESULTS

The experiments were conducted on a PC with an Intel i7 processor at 3.4 GHz, 8 GB of RAM and a CUDA-enabled NVIDIA GeForce GTX560 graphics card. The employed dataset was a collection of 15 videos from three different categories: i) 151 min. of news shows from the German public broadcaster RBB[1], ii) 140 min. from a cultural heritage show of the Dutch public broadcaster AVRO[2], called "Antiques Roadshow", and iii) 140 min. of videos from the archive of the Netherlands Institute for Sound and Vision[3].

Table 1. Experimental results for the considered techniques.

Technique	Precision	Recall	F-Score
T_0	0.887	0.917	0.902
T_1	0.891	0.915	0.903
T_2 [9]	0.860	0.906	0.882
T_3 [12]	0.745	0.469	0.576
T_4 [12]	0.919	0.377	0.535

The performance of the proposed GPU-based approach (denoted T_0 in the sequel), concerning both the detection accuracy and the time efficiency, was evaluated and compared against the performance of the following four CPU-based techniques: i) a variation of the proposed algorithm that uses exclusively CPU processing, denoted T_1, ii) the method of [9], which employs global visual features in combination with SVM classifiers, denoted T_2, iii) a technique similar to the method proposed in [12], using SIFT descriptors, dense sampling and a predefined codebook of 1K visual words, denoted T_3, and iv) an extension of the previous technique, also presented in [12], combining SIFT descriptors, dense sampling, a predefined codebook of 1K visual words and RGB color histograms, denoted T_4.

Ground-truth segmentation of the employed video dataset was created by human annotation of the shot boundaries. Overall, our dataset contains 3647 shot transitions, where 3216 of them are abrupt and the remaining 431 are gradual. For each one of the tested approaches we counted the number of correct detections, misdetections and false alarms and expressed them in terms of Precision, Recall and F-Score. Time efficiency was evaluated by expressing the required processing time as a factor of real-time processing, i.e. comparing these times with the actual duration of the processed videos (a factor below 1 indicates faster-than-real-time processing).

The experimental results are summarized in Table 1. As can be seen both the CPU- and the GPU-based implementation of the proposed algorithm perform considerably well, achieving high levels of precision and recall, which are slightly better than the corresponding results of the SVM-based technique of [9]. Moreover, they clearly outperform the other tested methods that rely on the use of SIFT local descriptors. As shown in Table 1, these techniques exhibit similar precision scores for significantly lower recall scores, which is consistent with the findings of [12].

Regarding the time performance, the running time of the proposed algorithm is about 0.3 of the video's duration, whereas the corresponding time for the CPU-based implementation and the algorithm in [9] were 0.88 and 1.25, respectively. The difference between the last two approaches

[1]http://www.rbb-online.de

[2]http://avro.nl

[3]http://www.beeldengeluid.nl

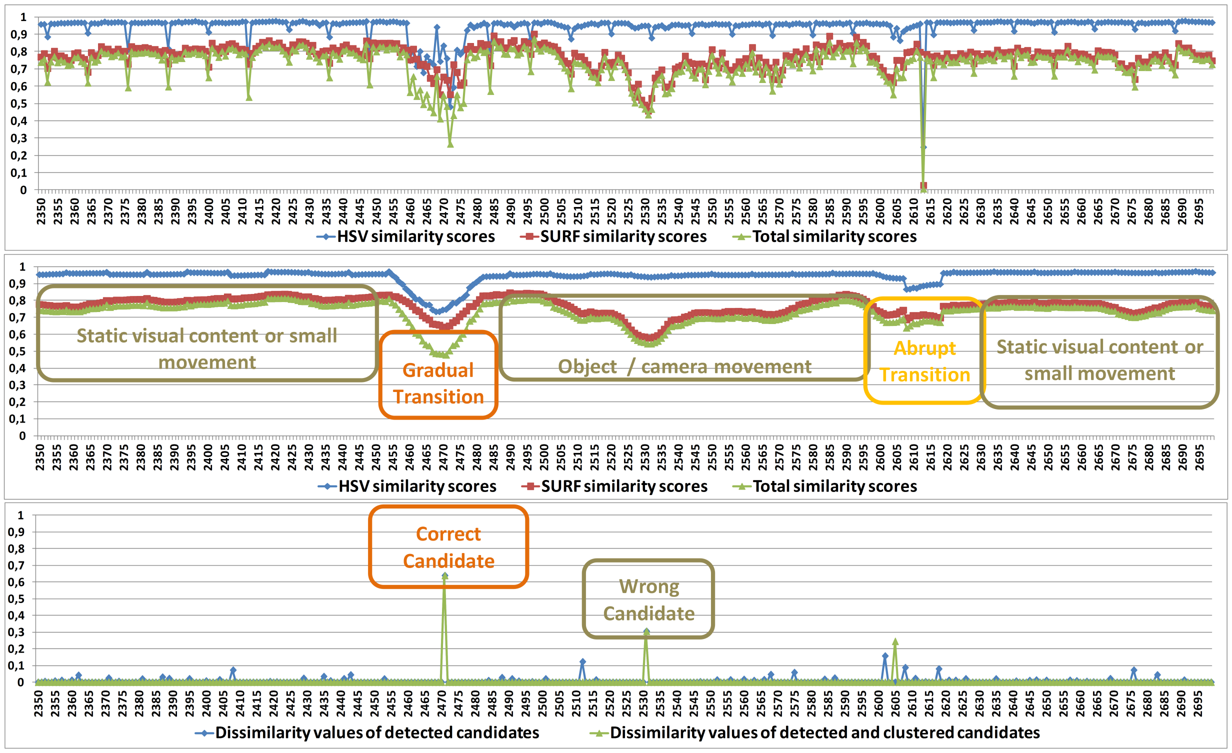

Fig. 1. Analysis of the similarity scores for the detection of gradual transitions. The top graph illustrates the frame similarity scores calculated by HSV histogram comparison (first fraction in Eq. (1)), SURF descriptor matching (second fraction in Eq. (1)) and their combination (**F**), for 350 frames of a video. The middle graph presents the computed moving average (curve **G**) for each of these score curves. The bottom graph shows the calculated dissimilarity vectors before (**D**) and after (**C**) clustering.

Table 2. Time efficiency of the CPU- and the GPU-based versions of the proposed algorithm.

Processing Step	% of Video Duration	
	CPU	GPU
HSV histogram extraction & comparison (Sec. 3.1)	2.49%	2.64%
SURF descriptor extraction & matching (Sec. 3.1)	84.02%	26.04%
Remaining processes of Sec. 3.1	0.91%	0.89%
Gradual transition detection (Sec. 3.2)	<0.01%	<0.01%
Identifying and filtering outliers (Sec. 3.3)	0.63%	0.34%
Total time	88.05%	29.91%

is reasonable since the method in [9] employs visual features that introduce higher computational complexity, compared to the calculation of HSV histograms and SURF descriptors, while the detection of shot boundaries via SVM classifiers is more time consuming compared to simple tests with predefined thresholds. Moreover, using GPU computing in the most computationally intensive parts of the algorithm we achieve significant gains compared to the CPU-based implementation, as shown in Table 2 (3x faster-than-real-time processing). Concerning the other methods, the reported time in [12] is half of the video's frame-rate, which means twice the video's duration. This is explained by the fact that the calculation of SIFT descriptors over a dense pixel-grid and their assignment to a codebook, as in [12], require significant amounts of computations.

5. CONCLUSIONS

In this paper we presented a method for the fast segmentation of videos into shots. The proposed algorithm employs HSV histograms and SURF descriptors for representing the visual content of the video frames, and detects abrupt and gradual transitions by quantifying and comparing the dissimilarity between pairs of frames against predefined thresholds. A refinement step removes outliers due to object/camera movement and flash-lights, further improving the detection accuracy. Aiming at real-time performance, we accelerated parts of the algorithm with GPU-based processing. The experiments show that the proposed technique can run about 3 times faster than real-time, and achieves very accurate video shot segmentation.

6. REFERENCES

[1] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, Jan. 1993.

[2] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying production effects," *Multimedia Systems*, vol. 7, no. 2, pp. 119–128, Mar. 1999.

[3] Z. Qu, Y. Liu, L. Ren, Y. Chen, and R. Zheng, "A method of shot detection based on color and edge features," in *1st IEEE Symposium on Web Society (SWS)*, 2009, pp. 1–4.

[4] L. Liu and J.-X. Li, "A novel shot segmentation algorithm based on motion edge feature," in *2010 Symposium on Photonics and Optoelectronic (SOPO)*, 2010, pp. 1–5.

[5] V. N. Vapnik, *The nature of statistical learning theory*, Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[6] V. Chasanis, A. Likas, and N. Galatsanos, "Simultaneous detection of abrupt cuts and dissolves in videos using support vector machines," *Pattern Recogition Letters*, vol. 30, no. 1, pp. 55–65, Jan. 2009.

[7] X. Ling, O. Yuanxin, L. Huan, and X. Zhang, "A method for fast shot boundary detection based on svm," in *Congress on Image and Signal Processing (CISP)*, 2008, vol. 2, pp. 445–449.

[8] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the 4th ACM international conference on Multimedia*, New York, NY, USA, 1996, MULTIMEDIA '96, pp. 65–73, ACM.

[9] E. Tsamoura, V. Mezaris, and I. Kompatsiaris, "Gradual transition detection using color coherence and other criteria in a video shot meta-segmentation framework," in *Workshop on Multimedia Information Retrieval at the 15th IEEE International Conference on Image Processing (ICIP)*, 2008, pp. 45–48.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[12] J. Lankinen and J.-K. Kämäräinen, "Video shot boundary detection using visual bag-of-words," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, Barcelona, Spain, 2013.

[13] J. Li, Y. Ding, Y. Shi, and W. Li, "A divide-and-rule scheme for shot boundary detection based on sift," *Journal of Digital Content Technology and its Applications*, pp. 202–214, 2010.

[14] J. Baber, N. Afzulpurkar, M.N. Dailey, and M. Bakhtyar, "Shot boundary detection from videos using entropy and local descriptor," in *17th International Conference on Digital Signal Processing (DSP)*, 2011, pp. 1–6.