

# Detection of Global Story Units in Full-length Movies

## Brief summary

One of the main steps in development of new user-friendly search concepts is finding a suitable way of organizing large amounts of stored video material before presenting it to the user for browsing purposes. Also, algorithms are needed to perform such video content organization automatically.

Most approaches proposed so far, organize video material on the base of individual video shots. However, since human understanding of a video content is far more event-based, rather than being in terms of single shots, event-based video abstraction for browsing is required. Such abstraction should, especially in case of long video sequences, result in a transparent *event cluster tree*, having single events as fundamental retrieval units on its lowest level and more complex story units as event collections on higher tree levels.

In this paper we propose a way of automatically obtaining highest levels of such a tree for a full-length movie, by detecting its *global story units*.

## Contact author:

Alan Hanjalic  
Delft University of Technology, Department of Electrical Engineering  
P.O. Box 5031, 2600 GA Delft, The Netherlands  
Phone: ++31-15-2783084, Fax: ++31-15-2781843  
e-mail: alan@it.et.tudelft.nl

# Detection of Global Story Units in Full-length Movies

Alan Hanjalic, Reginald L. Lagendijk, Jan Biemond

Delft University of Technology, Department of Electrical Engineering  
P.O. Box 5031, 2600 GA Delft, The Netherlands  
Phone: ++31-15-2783084, Fax: ++31-15-2781843  
e-mail: {alan,inald,biemond}@it.et.tudelft.nl

**Keywords:** Video content analysis and organization, video browsing, video databases

# Detection of Global Story Units in Full-length Movies\*

**Keywords:** Video content analysis and organization, video browsing, video databases

## Abstract

In digital video libraries an important problem is the efficient tracing and retrieving of any part of stored video. With increasing capacity of digital storage media, the usage of standard VCR search functions becomes unpractical, which calls for development of new user-friendly search concepts. One of the main steps hereby is to find a suitable way of organizing large amounts of stored video material before presenting it to the user for browsing purposes. When an appropriate method is found, algorithms are needed to perform such video content organization automatically, since doing this manually is a tedious and complex task.

In most approaches in literature, video content organization is based on individual video shots, which are used as the smallest logical units of video. However, since human understanding of a video content is far more concentrated on events (pieces of actions, dialogs, specific scenes, etc.) and higher semantic units (collections of interrelated events), rather than on single shots, event-based video abstraction for browsing is required. Such abstraction should, especially in case of long video sequences, result in a transparent *event cluster tree*, having single events as fundamental retrieval units on its lowest level. In this paper we propose a way of automatically obtaining highest levels of such a tree in case of a full-length movie, by detecting its *global story units*. Each such unit contains several temporally consecutive events, which are logically interrelated and belong to the same larger part of the story.

## 1 Introduction

In the fast growing area of video databases, the problem to efficiently retrieve stored video material remains one of the key issues [8]. The efficiency of the retrieval procedure determines, at the same time, its user-friendliness. To reach a high efficiency level, special attention has to be given to development of suitable methods for abstracting

large amounts of video material and presenting it to the user in a logical, compact and transparent form.

Since for human understanding of a video content an *event* like action clip, dialog, specific indoor/outdoor scene, is the most logical retrieval unit, it should be used as the base for a browsing-friendly video content organization. This event-based organization can be realized in different ways. In case of longer sequences, building of an *event cluster tree* is highly recommended. Such a tree has single events as fundamental retrieval units at its lowest level. Each higher tree-level contains semantically interrelated events in form of larger story units. In this way, the structure of the entire sequence can be presented at highest tree-levels according to user's interest. We apply the idea of making an event cluster tree on a full-length movie. In this paper we discuss all advantages of such an organization having in view the chosen "movie" application (section 3) and propose an algorithm for automated detection of *global story units* (further referred to as GSU), which can be used as the highest level of the event cluster tree and the starting point for building the whole tree until the lowest level, by performing more and more detailed content analysis within each GSU (section 4). Section 5 shows the results of applying the detection algorithm on a sample test sequence, which is followed in section 6 by discussions and prospects for our future work.

The mentioned aspect of automation in the process of detecting GSU boundaries is particularly important due to the following reasons:

- Analyzing the movie content manually and making the content abstraction in a browsing-friendly form is a tedious and complex task
- Service providers are generally not willing to send all necessary information needed for browsing. For instance, in case of streams belonging to the DVB standard, only

---

\* This work was supported in part by the EU ACTS program under the contract AC018: SMASH (Storage for Multimedia Applications Systems in the Home).

general information about the program is available (DVB - Service Information, [7]).

The specific “movie” application in this paper is chosen in relation to our activities within the SMASH project [9]. This project focusses on the development of the new-generation digital home storage device, and has the application related to movies as one of the main target applications. Especially in case of such home storage systems, designed to provide the maximum of user-friendliness, the necessity to automate all steps prior to the actual user-interaction becomes large after taking into account the two items related to the complexity of making a video abstract and the attitude of service providers.

## 2 Existing approaches for video content organization and related problems

The main problem in detecting higher logical units of video is how to transfer human high-level semantic meaning/interpretation about a particular video content to the machine level, where the entire content analysis, corresponding to this interpretation, is performed using low-level visual and temporal features like color, texture, shape, motion, etc.

Algorithms developed so far usually perform necessary image/video analysis and processing steps only to obtain some lower semantic elements, like video shots. This shot detection is mostly followed by grouping shots with similar contents, and building shot-based organization structures (e.g. shot cluster trees). There are also proposals to simulate the story flow by connecting different shot clusters in a suitable way (scene transition graphs in [2]). Problems with such approaches are that the obtained structure might be very large due to a huge number of shots in a full-length movie, and not transparent enough if no natural cluster structure exists among video shots. Performing the clustering procedure on individual video shots also does not generally provide the desirable event-based video organization, since shots from different parts of a sequence are clustered together not because they belong to one and the same event, but according to their individual visual and temporal characteristics.

One possibility for detecting larger logical units is to compare shots along larger time intervals. Using this idea, a method is proposed in [1] for detecting specific events, and smaller story units by investigating interrelationship of temporally consecutive video shots.

## 3 Global story units in a full-length movie

In this paper we also investigate inter-shot dependencies, but in a different way and for other purposes. It is our intention to recognize the global structure of a typical Hollywood-made full-length movie. This global structure is obtained by detecting *global story units* (GSU) in the movie material.

The inspiring idea for the work presented here, came by analyzing the problem of obtaining a transparent, event-based video content organization in the video-browser, leading to satisfaction of a wide range of users. One possibility would be to first detect all events in a movie and then apply a clustering algorithm to obtain higher logical levels. We see two problems by this approach: (1) methods for detecting individual events are not able to capture the entire movie, (2) if events belonging to different movie parts are clustered together, the transparency of higher tree levels can not be guaranteed. Therefore, the idea of building an event cluster tree in the opposite direction seems to be much more attractive. This is done by detecting more global logical units first and then performing more detailed event-oriented analysis within each unit.

- 1 Movie introduction
- 2 Running to the wedding
- 3 Wedding I
- 4 In the park after the wedding
- 5 Reception in the tend
- 6 In the pub, downstairs
- 7 In the room, night
- 8 In the room, morning
- 9 Running to the wedding
- 10 Wedding II
- 11 Reception in the tend
- 12 Hotel room after the wedding
- 13 Shopping for the wedding
- 14 Bridal shop
- 15 Cafe
- 16 In the shopping mall
- 17 On the street
- 18 Wedding III
- 19 Funeral
- 20 Wedding IV
- 21 At home, after a “disaster”
- 22 New marriage proposal

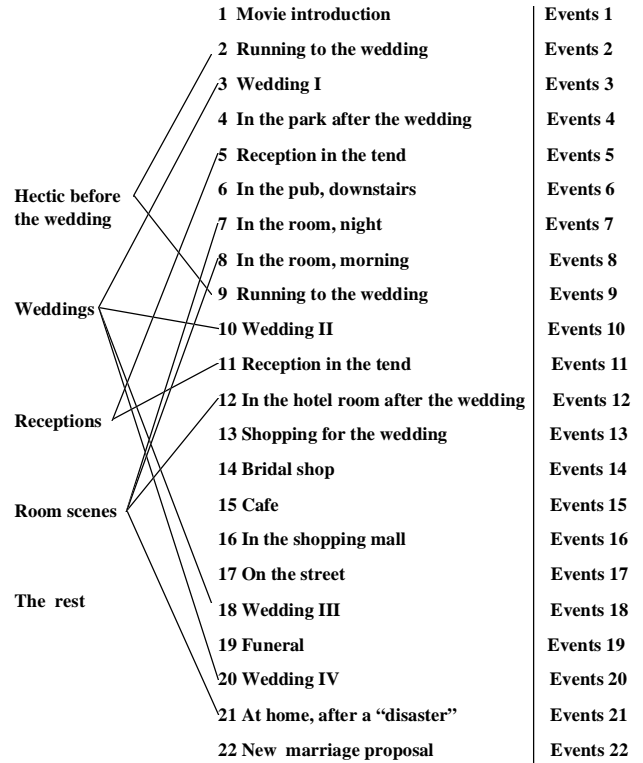
**Table 1:** Representation of the entire movie using global story units

Before describing our approach for automated detection of GSU boundaries, we first explain the concept and purpose of GSUs using a sample sequence (“Four Weddings and a Funeral”, courtesy of Polygram).

We define *global story units (GSU)* as collections of temporally consecutive, and semantically interrelated *events*. The whole movie can be understood as a series of consecutive GSUs. What is meant by the term GSU, is illustrated by the Table 1, where the content of the entire movie is represented in terms of these units. 22 GSUs from Table 1 are the result of author’s own movie analysis. While here each GSU is given a textual label to describe its content, suitable visual means (frames or pieces of video) will be used in the practical case. The first impression from Table 1 is that such a representation is surprisingly compact (having in mind the movie length), but still reveals clearly what can be expected in each of selected GSU-items. All events belonging to one GSU are in one way or another interrelated, leading to the semantic unity of the entire GSU. One example is the GSU 11, showing the reception after the wedding. It is composed of a large number of small dialogs and camera shots showing different people in different activities (eating, dancing, talking, etc.).

After defining GSUs in the way it is done in the table, the possibility should be used (if any) to group GSUs with similar contents in larger clusters. This step can be performed after defining all GSUs and by comparing them using their individual characteristics. In the movie example here, all 4 wedding GSUs can be put together, also two GSUs of running to the wedding and two big reception GSUs. The same can be done with several indoor GSUs, since e.g. the scenery is similar in each case. Such clustering procedure provides even better entrance level in the browsing interface. As shown in Table 2, the first level of the video-browser contains 5 clusters giving the user a clear idea about the movie structure. To perform further content analysis within a GSU, specialized event-detecting approaches can be used, such as one proposed in [1]. After being detected, all events belonging to one GSU can then be reached directly, or a cluster-tree of events can be made within each GSU using certain criteria for events comparison.

Table 2 shows events-based tree-organization of the movie with 3 levels: (1) clustered GSUs (this might be especially useful in case that a movie follows two stories in parallel, so that e.g. from each two consecutive GSUs one belongs to one story and one to the other), (2) GSUs and (3) Events-level (all events belonging to a GSU in a plain or clustered organization).



**Table 2:** Event-based organization of the full-length movie “Four Weddings and a Funeral”

#### 4 Automation of GSU boundaries detection

As already mentioned in the introduction, it is necessary to perform all steps of building the tree-structure of the movie content organization in an automated way, especially in case of home video storage devices (SMASH).

Each GSU, as defined in the previous section, represents a higher-level semantic unit. Therefore the problem to practically implement the algorithm for detecting GSU boundaries, using only low-level visual features, is not easy to solve. We propose an algorithm, based on analysis of interrelations among consecutive video shots. The idea behind it, is that one GSU consists of several consecutive events, each containing a certain number of interrelated shots. Since all events within a GSU are also interrelated in some way, this can also be expected for all shots within that GSU. Interrelations among shots are obtained using their specific visual characteristics, which are collected from shot key frames.

**Global story unit 13: Shopping for the wedding**



**Global story unit 14: Bridal shop**



**Global story unit 15: Cafe**



**Global story unit 16: In the shopping mall**



**Global story unit 17: On the street**



**Global story unit 18: Wedding III (Scotland)**



**Figure 1:** Example of six consecutive GSUs, capturing almost 45000 frames (approx. 30 min.) of the movie. Each frame represents one video shot of a GSU

**4.1 Video shots and key frames**

Video shots are defined as continuous camera drives, lasting several seconds and containing one and the same video material. Boundaries between consecutive shots are detected by comparing the contents of consecutive frames

in the video sequence. Here, the frame content is typically measured by features such as color, texture and shape.

After a shot change (abrupt or gradual), a new shot starts, containing either the same material as the previous one but taken in a different way, or something completely different.

For characterizing the shot content, specific frames (*key frames*) are used very frequently. If they are located optimally, they eliminate temporal redundancy among shot frames and capture all shot-characteristic visual features in a compact form.

#### 4.2 Algorithm

We first obtain individual video shots, by employing the approach from [3], which uses locally computed threshold based on a statistical model for frame-to-frame differences. In the next step, each shot is represented by a number of key frames. In our tests we used only two key frames per shot, although we believe that the performance of the algorithm can be improved using more sophisticated key frame extraction methods, as proposed in [4], [5] or [6].

For each shot  $k$ , the *GSU boundary likelihood*  $L_k$  is computed, by measuring its interrelationship to next-following shots.  $L_k$  represents the likelihood that the boundary between two consecutive GSUs can be assumed to lie around the shot  $k$ .

In order to obtain a realistic picture about this interrelation, checking  $N$  next-following shots ( $N > 1$ ) is necessary. We explain this by the fact that recognizing higher logical units of video is possible only by investigating video material over longer time intervals. If the tested shot is not at the GSU boundary, it can be expected that at least one of  $N$  next-following shots belonging to the same GSU is similar to the tested shot. If the tested shot lies at the GSU boundary, all  $N$  next-following shots belong to a different GSU, characterized e.g. by a different scenery, and being highly dissimilar to the tested shot. For the likelihood value  $L_k$ , the best match (minimum dissimilarity) out of  $N$  shot-comparisons is taken, assuming that a suitable comparison criterion is found. If several comparison criteria are used independently, minimum dissimilarities for each criterion are multiplied for obtaining the likelihood value. Such increase of number of criteria for shot comparison proved to give better results, compared with only one criterion. Besides of making the decision more reliable (if one criterion for some reason leads to false conclusion, the others can correct it), the multiplication of dissimilarities leads to better recognizable detection peaks in the likelihood curve. This curve is obtained after drawing all likelihood values over the axis containing video shot indices (Figure 3). Used comparison criteria should reveal only global visual structure of shots, which are also characteristic for the whole GSU. In our experiments we used two criteria:

- Global luminance tendency in the shot, obtained by computing a 2-bin histogram (capturing gray values 0-127, 128-255) over all key frames in the shot.
- Color composition of the shot-background (Figure 2), obtained by computing color histogram with 64 colors (bins) using 4x4x4 resolution of the RGB space, over all key frames in the shot.

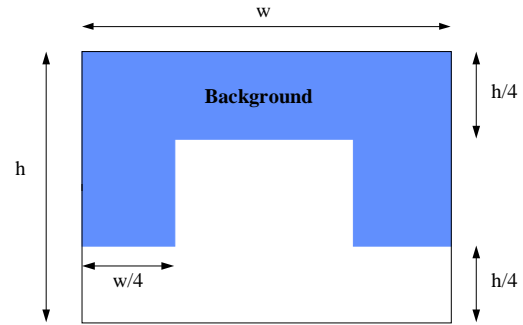
Taking into account the entire previous discussion,  $L_k$  is now computed as follows:

$$L_k = \min_{i=1,N} d_{lum}(k, k+i) * \min_{i=1,N} d_{bgd\_col}(k, k+i)$$

Dissimilarity between two shots  $k$  and  $k+i$ , in relation to each of these criteria, is computed using the *standard mean absolute difference* equation for comparing histograms:

$$d_{criterion}(k, k+i) = \sum_x |h_k^{criterion}(x) - h_{k+i}^{criterion}(x)|$$

with  $h_k^{criterion}(x)$  being the overall histogram of the shot  $k$  computed according to the chosen *criterion* and  $x$  going over all bins (number of bins also defined by the criterion).



**Figure 2:** Part of a key frame used for computing the color histogram of the shot-background

The likelihood curve as shown in Figure 3, if obtained for the whole movie, reveals the whole movie structure and leaves the possibility to classify the *entire* movie material in terms of GSU. How good and easy GSUs can be detected, depends on how distinguishable GSU boundary peaks are, compared to other non-boundary likelihood values. Possible problems in recognizing boundary peaks on the likelihood curve decrease with increasing quality

and number of criteria for computing likelihood values. For automated boundary peak detection, various thresholding techniques known from the shot change detection (video parsing) can here also be applied (e.g. [3]).

## 5 Results

For evaluating the proposed approach, a 30 minutes part of the movie “Four Weddings and a Funeral” has been used. This part of the movie contains GSUs 13-18 (Figure 1) and 184 detected shots.

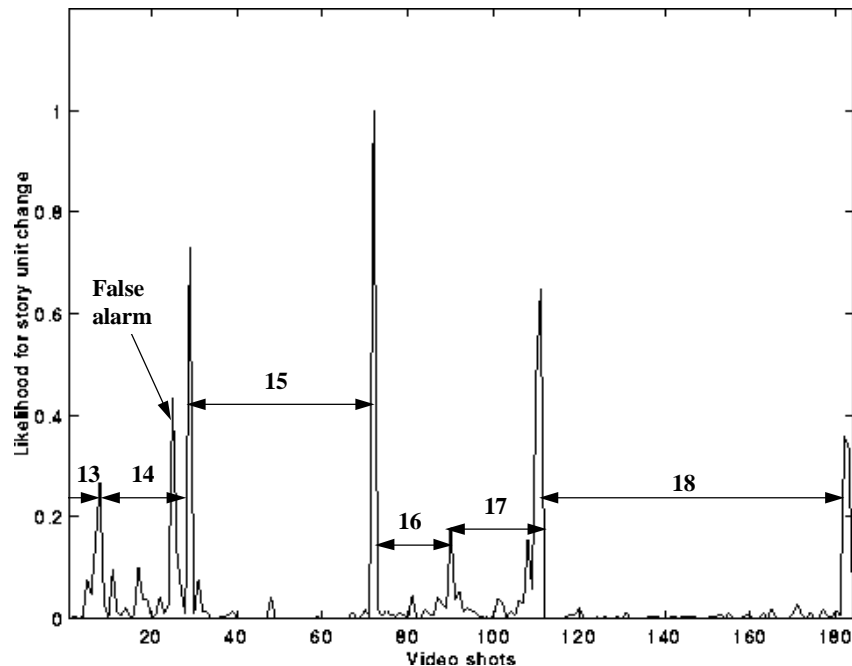
Some shot changes in the first algorithm step were missed, but this has not influenced test results. Such mistakes can eventually cause false alarms in detecting GSU changes, since in these cases visual features belonging to several shots are merged together and used for comparisons with other shots. However, we believe that the probability for a false alarm due to this reason is relatively small. We can explain this by the fact, that a missed shot change occurs most likely between shots with similar visual contents. This means that merging visual features of these two shots does not provide false information for the actual GSU detection procedure.

The likelihood value has been computed for each shot, as described in the previous section, using  $N=5$ . The resulting likelihood curve can be seen in Figure 3. Sharp peaks showing all GSU boundaries in the used sequence can be clearly recognized.

## 6 Discussion

In this paper we discussed all advantages of organizing the content material of a full-length movie by building an event cluster tree. We also proposed a method for automatically obtaining highest levels of such tree. This is done by detecting consecutive global story units in a movie, each containing several interrelated events. In this way, the entire movie material is captured. Using some algorithms proposed in literature (e.g. [1]) but also the here proposed algorithm, it is than possible to perform more detailed event-based content analysis within each of defined GSU.

The proposed algorithm for automated detection of GSU boundaries performed well on our test sequence (Figure 3). Our coming activities will be focused on testing the algorithm on broader scope of sequences and on making it more robust, e.g. by using more than two criteria for measuring the GSU boundary likelihood.



**Figure 3:** Results of the sequence analysis. The same 30 minutes of the sequence have been examined as represented by GSUs in Figure 1. Peaks show the boundaries between two consecutive global story units.

## References

- [1] Yeung M., Yeo B.-L.: “*Video Content Characterization and Compaction for Digital Library Applications*”, In the Proceedings, IS&T/SPIE Storage and Retrieval for Image and Video Databases V, Vol. 3022, February 1997
- [2] Yeung M., Yeo B.-L., Wolf W., Liu B.: “*Video Browsing using Clustering and Scene Transitions on Compressed Sequences*”, In the Proceedings, IS&T/SPIE Multimedia Computing and Networking, February 1995
- [3] Hanjalic A., Ceccarelli M., Lagendijk R.L., Biemond J.: “*Automation of systems enabling search on stored video data*”, In the Proceedings, IS&T/SPIE Storage and Retrieval for Image and Video Databases V, Vol. 3022, February 1997
- [4] A. Hanjalic, R.L. Lagendijk, and J. Biemond, “A New Key-Frame Allocation Method for Representing Stored Video-Streams”, *to be presented in the First International Workshop on Image Databases and Multi Media Search*, Amsterdam, The Netherlands, 1996
- [5] Lagendijk, R.L., Hanjalic, A., Ceccarelli, M.P., Soletic, M., Persoon, E.: “Visual Search in a SMASH System”, In the Proceedings of IEEE International Conference on Image Processing 1996
- [6] Zhang H., Low C.Y., Smoliar S.W., “Video Parsing and Browsing using Compressed Data”, *Multimedia Tools and Applications*, vol. 1, pp. 89-111, Kluwer Academic Publishers, 1995.
- [7] ETS 300 468, "Digital broadcasting systems for television, sound and data services; specification for Service Information (SI) in Digital Video Broadcasting (DVB) systems", EBU/ETSI JTC, January 1996
- [8] Picard, R.W.: “Light-years from Lena: Video and Image Libraries of the Future”, In the Proceedings, IEEE International Conference on Image Processing 1995, Vol.1, pp. 310-313
- [9] The SMASH project home page:  
<http://www-it.et.tudelft.nl/pda/smash>