Systematic Evaluation of Logical Story Unit Segmentation

Jeroen Vendrig, *Member, IEEE,* and Marcel Worring

***Abstract*—Although various Logical Story Unit (LSU) segmentation methods based on visual content have been presented in literature, a common ground for comparison is missing. We present a systematic evaluation of the mutual dependencies of segmentation methods and their performances.**

LSUs are subjective and cannot be defined with full certainty. To limit subjectivity, we present definitions based on film theory. For evaluation, we introduce a method measuring the quality of a segmentation method and its economic impact rather than the amount of errors. Furthermore, the inherent complexity of the segmentation problem given a visual feature is measured. Also, we show to what extent LSU segmentation depends on the quality of shot boundary segmentation.

To understand LSU segmentation, we present a unifying framework classifying segmentation methods into four essentially different types. We present results of an evaluation of the four types under similar circumstances using an unprecedented amount of 20 hours of 17 complete videos in different genres. Tools and ground truths are available for interactive use via internet.

***Index Terms*—Video content analysis, video representation, video segmentation.**

I. Introduction

VIDEO structure is important for abstracting, visualizing, searching, and navigating through videos. Brunelli [1] defines video structure as the decomposition of a stream into shots ("contiguously recorded sequences") and scenes. In video processing literature, the term scene is already occupied and renamed to Logical Story Unit (LSU) [2] or simply story unit [3], [4]. In cinematography an LSU is defined as *a series of shots that communicate a unified action with a common locale and time* [5]. Viewers perceive the meaning of a video at the level of LSUs [3], [6]. Therefore, next to methods for accurate shot detection there is even a greater need to have methods for automatic segmentation of a video into LSUs [2], [4], [6]–[16].

Large-scale evaluations on shot segmentation results have been performed already, e.g., [1], showing reasonable performance. Such a major effort has not been done before for LSU segmentation results. To our knowledge, the largest data set described in literature is four hours for just two movies [2]. Other authors use more videos, but not full-length [6], [12]. To evaluate LSU segmentation results, a set of complete videos from various genres should be used.

Manuscript received March 12, 2002. This work was supported by NWO/SION under Grant 612-61-005 and by the Multimedia Information Analysis project. The associate editor coordinating the review of this paper and approving it for publication was Dr. Hong-Jiang Zhang.

The authors are with the Computer Science Institute, University of Amsterdam, 1098 SJ Amsterdam, The Netherlands (e-mail: vendrig@science.uva.nl).

Digital Object Identifier 10.1109/TMM.2002.802021

Methods presented in literature are evaluated by their creators, resulting in three major problems. First, there is no common data set for evaluation. Second, the definition used for an LSU is not precise. Therefore, definitions of a ground truth depend on individual viewers instructed by the creator of a method. The ground truth may be biased to the specific method's underlying assumptions and approach. Third, for each method specific features are used to compare video content. Hence, it is possible that the quality of features rather than methods is evaluated. What is required is an independent evaluation.

In literature, evaluation of segmentation results is either left to the reader [4] or based on evaluation criteria for shot boundary segmentation. The latter boils down to counting false negatives and false positives [2], [6]. This approach requires an exact, unbiased ground truth. Counting just the amount of errors does not communicate error magnitude, i.e., the economic impact errors have on the result. For shots, just counting is feasible, because start and end are well defined. For LSU ground truths, this cannot be expected.

In the context of this paper, we restrict ourselves to generally applicable segmentation methods using visual features. In contrast, edits-based analysis requires an explicit model of a video program [17] or a set of generic style rules used in editing. Editing rules have been applied for specific classes of feature films [9] only. Segmentation methods that use other modalities, such as audio and text [18], yield partial information only. They still depend on visual information. Evaluation of visual methods is necessary in finding the best starting point.

This paper is organized as follows. In Section II, we present a new, more precise definition for LSU. Assumptions and techniques underlying LSU segmentation methods are made explicit, resulting in a unifying framework. In Section III, user centered measures for visual features and segmentation methods are defined. In Section IV, results of the performance evaluation are described. Conclusions are given in Section V.

II. Logical Story Unit Segmentation

A. Consistent LSU Definition

In this section we define an LSU such that it can be applied consistently to a large video collection consisting of movies and TV series.

Since humans perceive LSUs by way of changes in content [3], an LSU can be defined best by its boundaries.

Definition 1: An LSU is the series of shots between an LSU boundary and the next LSU boundary.

This definition allows us to reformulate the segmentation problem into finding discontinuities in place or time. At first

sight, this seems a trivial task. In practice, however, the level of detail to be used in defining a discontinuity depends on the content of the movie and hence human interpretation.

Consistently defining the LSU concept requires guidelines for cases where subjectivity plays a role. Since movies are the result of an artistic process, sometimes causing confusion on purpose, the necessity for human judgment will always remain to some extent. However, we provide guidelines for four important and often used editorial techniques, as identified in film literature [5], [19]. The guidelines ensure that most cases of subjectivity are caught.

The four editorial techniques and related guidelines for consistent LSU definition are as follows.

- *Elliptical editing*, in which "an action is presented in such a way that it consumes less time on the screen than it does in the story" [19].

  As viewers perceive the shots as being continuous in time, the shots should be considered part of the same LSU.
- *Montage*, the juxtaposition of shots based on a common theme.

  When the shots cannot be perceived as having a common locale or time, the shots should not be considered belonging to the same LSU.
- *Establishing shot*, "a beginning shot of a new scene that shows an overall view of the new setting" [5].

  Since often the establishing shot shows the outside of the building where a LSU takes place, the "common locale" part of the LSU definition is interpreted broadly. Thus, an establishing shot should be part of the LSU for which it determines the setting.
- *Parallel cutting*, the alternation of shots at different locales to create the impression that several events take place at the same time [5].

The definition of an LSU as given in the introduction does not accommodate parallel cutting. We let the part of the definition that an LSU is a series of shots prevail. Hence, events shown in parallel cutting should be considered as one LSU.

Determining when a sequence of shots is parallel cutting, is a subjective task in itself. Therefore, we define parallel cutting more exact. When a discontinuity is a small interruption, i.e., the story later continues in the same locale and time, this is attributed to parallel cutting. To this end, we introduce a maximal gap $g_{\max}$ between discontinuities. Parallel cutting is continuity in time or locale between two shots that are not immediately succeeding one another, but that are no farther apart than the maximal gap $g_{\max}$.

Let us formalize the concept of LSU boundary, accommodating the problem of parallel cutting. The series of shots in a video is denoted by V_σ. A shot in V_σ is represented by σ_x, where x is the shot index number. The continuity operator $c(\sigma_x, \sigma_y)$ returns *true* if σ_x and σ_y share time and locale and σ_x and σ_y are both part of V_σ, and returns *false* otherwise.

Definition 2: σ_x is an LSU boundary, if

$$c(\sigma_{x-1}, \sigma_x) = \text{false} \wedge \{\forall_{y,z}: \quad c(\sigma_y, \sigma_z) = \text{false} \wedge (y < x < z) \wedge (z - y - 1 \leq g_{\max})\}$$

Variable $g_{\max}$ should be chosen once for the video collection to be segmented. Similar to [9], we have found $g_{\max} = 3$ shots to be a representative value.

Given the definition and representation of an LSU, the process of detection of LSUs and evaluation of detected LSUs can be described.

B. Assumptions for Use of Visual Similarity

In this section, we describe general problems and assumptions underlying the broad class of segmentation methods based on visual similarity [2], [4], [6]–[16].

A problem for LSU segmentation using visual similarity is that it seems to conflict with Definition 2, which is based on the semantic notion of common locale and time. There is no one-to-one mapping between the semantic concepts and the data-driven visual similarity. In practice, however, most LSU boundaries coincide with a change of locale, causing a change in the visual content of the shots. Furthermore, usually the scenery in which an LSU takes place does not change significantly, or foreground objects will appear in several shots, e.g., talking heads in the case of a dialogue. Therefore, visual similarity provides a proper base for common locale.

There are two complicating factors regarding the use of visual similarity. Firstly, not all shots in an LSU need to be visually similar. For example, one can have a sudden close-up of a glass of wine in the middle of a dinner conversation showing talking heads. This problem is addressed by the *overlapping links* approach [2] which assigns visually dissimilar shots to an LSU based on temporal constraints. Secondly, at a later point in the video, time and locale from one LSU can be repeated in another, not immediate succeeding LSU.

The two complicating factors apply to the entire field of LSU segmentation based on visual similarity. Consequently, an LSU segmentation method using visual similarity depends on the following three assumptions.

Assumption 1: The visual content in an LSU is dissimilar from the visual content in a succeeding LSU.

Assumption 2: Within an LSU, shots with similar visual content are repeated.

Assumption 3: If two shots σ_x and σ_y are visually similar and assigned to the same LSU, then all shots between σ_x and σ_y are part of this LSU.

For parts of a video where the assumptions are not met, segmentation results will be unpredictable.

C. Unifying Framework for Existing Methods

Given the assumptions, LSU segmentation methods using visual similarity can be characterized by two important components, viz. the shot distance measurement and the comparison method. The former determines the (dis)similarity mentioned in Assumptions 1 and 2. The latter component determines which shots are compared in finding LSU boundaries. Both components are described in more detail.

Shot Distance Measurement: The shot distance δ represents the dissimilarity between two shots and is measured by combining (typically multiplying) measurements for the *visual distance* δ^v and the *temporal distance* δ^t. The two distances will now be explained in detail.

TABLE I
CLASSIFICATION OF LSU SEGMENTATION METHODS

Comparison method	Temporal distance function	
	binary	continuous
sequential	overlapping links [2], [7], [8], [9], [10]	continuous video coherence [11], [12], [13]
clustering	time constrained clustering [4], [14], [15]	time adaptive grouping [6], [11], [16]

Visual distance measurement consists of dissimilarity function δ_f^v for a visual feature f measuring the distance between two shots. Usually a threshold τ_f^v is used to determine whether two shots are close or not. δ_f^v and τ_f^v have to be chosen such that the distance between shots in an LSU is small (Assumption 2), while the distance between shots in different LSUs is large (Assumption 1).

Segmentation methods in literature do not depend on specific features or dissimilarity functions, i.e., the features and dissimilarity functions are interchangeable amongst methods.

Temporal distance measurement consists of temporal distance function δ^t. As observed before, shots from not immediate succeeding LSUs can have similar content. Therefore, it is necessary to define a time window τ^t, determining what shots in a video are available for comparison. The value for τ^t, expressed in shots or frames, has to be chosen such that it resembles the length of an LSU. In practice, the value has to be estimated since LSUs vary in length.

Function δ^t is either binary or continuous. A binary δ^t results in 1 if two shots are less than τ^t shots or frames apart and ∞ otherwise [4]. A continuous δ^t reflects the distance between two shots more precisely. In [6], δ^t ranges from 0 to 1. As a consequence, the further two shots are apart in time, the closer the visual distance has to be to assigned them to the same LSU. Time window τ^t is still used to mark the point after which shots are considered dissimilar. Shot distance is then set to ∞ regardless of the visual distance.

The *comparison method* is the second important component of LSU segmentation methods. In *sequential iteration*, the distance between a shot and other shots is measured pair-wise. In *clustering*, shots are compared group-wise. Note that in the sequential approach still many comparisons can be made, but always of one pair of shots at the time.

Methods from literature can now be classified according to the framework. The visual distance function is not discriminatory, since it is interchangeable amongst methods. Therefore, the two discriminating dimensions for classification of methods are temporal distance function and comparison method. Their names in literature and references to methods are given in Table I. Note that in [11], two approaches are presented.

D. Discussion

As any binary function can be expressed as the limit case of some continuous function, methods using binary temporal distance can be considered special cases of the methods using continuous temporal distance. The binary function is more sensitive to the choice of τ^t than a continuous function. A larger value for τ^t allows for more shot comparisons resulting in less oversegmentation. The disadvantage of a binary function is that more shot comparisons also increase the number of shot pairs that are determined visually similar, resulting in undersegmentation, especially in combination with Assumption 3. Making τ^v more strict will not solve this problem. It results in oversegmentation since Assumption 2 is then easily violated. A continuous function does not suffer as much from the threshold setting dilemma, since shots with a higher temporal distance have to compensate with a lower visual distance.

The comparison method has a similar effect since it influences time window τ^t as well. Sequential comparison is sensitive to violation of Assumption 2 because it makes only shot-wise comparisons. τ^t and τ^v have to be fine-tuned for each video in order to achieve good results. Cluster comparison suffers less from the parameter setting problem, since shots are compared group-wise. This allows for a larger value of τ^t, because similarity of one pair of shots alone will not result in a new LSU boundary. Furthermore, it allows for a more strict value of τ^v. Similarity is measured in a group of shots and therefore less sensitive to outliers.

III. EVALUATION

In this section, we present evaluation methods for features and segmentation methods. Evaluation is done from a video librarian point of view, reflecting the practical and economical effort required to correct errors.

A. Feature Evaluation

Humans and automated segmentation methods have different ways to find LSU boundaries based on discontinuities in time and locale. Humans try to relate changes in time and locale to discontinuities in meaning [3]. Automated methods depend on visual dissimilarity in the video content, as expressed in Assumptions 1 and 2. The semantic gap between human defined LSUs and so-called "computable scenes" [12] makes it impossible for automated segmentation methods to achieve fully correct segmentation based on visual features only. In this section, we present two criteria to measure to what degree automatic segmentation can approach human defined LSUs using a visual feature.

Measurement of the potential of a visual feature in segmenting a video into LSUs requires to measure the extent to which the Assumptions 1 and 2 hold. Measurements have to be taken in the context of Assumption 3 which allows shots to be assigned to LSUs regardless of the values for the visual feature.

To allow for a formal description of the two measurement criteria, we first define V_λ and $\breve{V}_\lambda$. Similar to shot segmentation V_σ, V_λ denotes a ground truth series of LSUs for a video, i.e., the desired segmentation result. The series of computable LSUs are denoted by $\breve{V}_\lambda$. For later use, we introduce the first shot and last shot operators FS and LS [4], returning the index of the shot.

Coverage $\mathcal{C}$ measures to what extent Assumption 2 is met in the ground truth, i.e., what part of the ground truth LSU λ_t could theoretically be found given the feature and visual threshold τ^v. To be precise, $\mathcal{C}$ is the fraction of shots in λ_t that can be

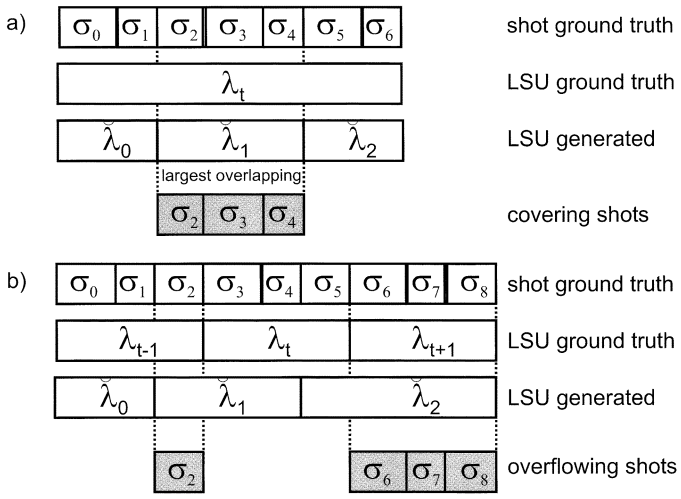

Fig. 1. Visualization of (a) coverage $\mathcal{C}$ and (b) overflow $\mathcal{O}$ for ground truth LSU λ_t. The gray shaded areas contribute to the value measured.

merged into a newly generated LSU $\breve{\lambda}_x$ [see Fig. 1(a)]. In the best case $\mathcal{C} = 1$, i.e., $\lambda_t = \breve{\lambda}_0$. Otherwise, there are several LSUs $\breve{\lambda}_0 \ldots \breve{\lambda}_n$, in which case the longest $\breve{\lambda}_j$ is taken to measure $\mathcal{C}$.

$$\mathcal{C}(\lambda_t) = \frac{\max_{j=0\ldots n} \#(\breve{\lambda}_j)}{\#(\lambda_t)} \tag{1}$$

where the $\#$ operator counts the amount of shots a representation of V or an LSU contains.

Overflow $\mathcal{O}$ measures to what extent Assumption 1 is met. It quantifies the overlap of a given λ_t with its two surrounding LSUs (λ_{t-1} and λ_{t+1}). $\mathcal{O}$ is the fraction of shots in the surrounding LSUs that would be merged with λ_t into a newly generated LSU $\breve{\lambda}$ [see Fig. 1(b)]. In the best case, $\breve{\lambda}_j = \lambda_t$ hence $\mathcal{O} = 0$. In the worst case, all three LSUs are merged into one $\breve{\lambda}_0$ and $\mathcal{O} = 1$.

$$\mathcal{O}(\lambda_t) = \frac{\sum_{j=0}^{n} \#(\breve{\lambda}_j \setminus \lambda_t) \cdot \min(1, \#(\breve{\lambda}_j \cap \lambda_t))}{\#(\lambda_{t-1}) + \#(\lambda_{t+1})}. \tag{2}$$

The measured values can be aggregated into values for an entire video or collection of videos as follows:

$$\mathcal{C}(V) = \sum_{t=0}^{\#(V_\lambda)-1} \mathcal{C}(\lambda_t) \cdot \frac{\#(\lambda_t)}{\#(V_\sigma)}. \tag{3}$$

$\mathcal{O}(V)$ is defined in a similar way.

There are three important applications for the measurements. First of all, they are useful to compare the performance of individual features. Second, the measurements show to what extent segmentation of a video sequence is theoretically possible, i.e., under ideal circumstances. The ideal feature/threshold combination has $\mathcal{C} = 1$ and $\mathcal{O} = 0$. The difference between the actual measurements and the ideal is the inherent complexity of the segmentation problem given the visual feature. Third, when coverage and overflow are plotted against one another, an appropriate threshold can be selected depending on the user's preferences for amount of overflow (undersegmentation) and coverage (oversegmentation).

1. **while** $\breve{V}_\lambda \neq V_\lambda$
2. Let $\breve{\lambda}_t$ be the first uncorrected LSU in $\breve{V}_\lambda$
3. **if** $c(\sigma_{FS(\breve{\lambda}_t)}, \sigma_{LS(\breve{\lambda}_t)}) = false$
4. Divide $\breve{\lambda}_t$ into λ_t and $\breve{\lambda}'_{t+1}$, by searching σ_x where $x = FS(\breve{\lambda}'_{t+1}) = LS(\lambda_t) + 1$
5. Update $\breve{V}_\lambda$, replacing $\breve{\lambda}_t$ with λ_t and $\breve{\lambda}'_{t+1}$
6. **else if** $c(\sigma_{LS(\breve{\lambda}_t)}, \sigma_{FS(\breve{\lambda}_{t+1})}) = true$
7. Search $\breve{\lambda}_x$ that contains $LS(\lambda_t)$
8. Divide $\breve{\lambda}_t \ldots \breve{\lambda}_x$ into λ_t and $\breve{\lambda}'_x$, where $\breve{\lambda}'_x = \sigma_{LS(\lambda_t)+1} \cdots \sigma_{LS(\breve{\lambda}_x)}$
9. Merge $FS(\breve{\lambda}_t) \ldots LS(\lambda_t) \in \breve{\lambda}_x$ into λ_t, $\breve{\lambda}'_{t+1} = FS(\breve{\lambda}_x) \ldots LS(\breve{\lambda}_x)$
10. Update similar to step 5

Fig. 2. User model procedure.

B. Evaluation Method

An evaluation criterion for the quality of an automatic LSU segmentation result should reflect the perception of users. In the case of LSUs, users have doubts about the exact start and end of an LSU, see for example the problem with establishing shots mentioned in Section II-A. An LSU evaluation criterion should therefore not measure *if* a boundary is incorrect but it should measure *how* incorrect the boundary is. Although it is impossible to completely solve the problem of biased ground truths, such criterion will at least cope with the uncertainty in truths rather than ignore it.

Similar to the video string edit distance proposed in [20] to measure similarity between video sequences, we propose to measure the cost of transforming result $\breve{V}_\lambda$ to the ground truth V_λ. This is done by counting the number of shot comparisons for continuity operator $c(\sigma_x, \sigma_y)$ necessary to correct LSU boundaries, as this is proportional to the practical effort to be delivered by video librarians. To that end, we introduce a simplified user model based on two assumptions:

- the user has a constant V_λ in mind;
- the user is able to carry out continuity operator c consistently, correctly and immediately.

The procedure the user follows to convert $\breve{V}_\lambda$ into V_λ is modeled as in Fig. 2. Basically, the user iterates over the found LSUs $\breve{\lambda}_t \in \breve{V}_\lambda$ and corrects them one at the time. In the end, each $\breve{\lambda}_t = \lambda_t$ and hence $\breve{V}_\lambda = V_\lambda$.

In the model, the user makes a number of assessments. He assesses whether the boundaries in $\breve{V}_\lambda$ are correct (lines 3 and 6). Note that this type of assessment is made even in the case of perfect segmentation $V_\lambda = \breve{V}_\lambda$. Then, if necessary, he makes assessments to find the true boundaries from V_λ in $\breve{V}_\lambda$ (lines 4 and 7). The amount of assessments to find the true boundaries depends on the search strategy of the user. A trivial strategy is a linear search, where the user simply iterates back or forward shot by shot. This is not realistic for an expert user, such as a video librarian. We use a more advanced version of the linear strategy. It is modeled as follows. The user first takes big steps forward or backward. We use steps of ten shots, corresponding to half the length of an average LSU. When the user realizes he has gone too far, he switches to small steps, we use steps of one shot, and iterates in the opposite direction.

Based on the user model, it is possible to define evaluation criteria that can be measured consistently for different automatic segmentation methods.

C. Evaluation Criteria

The quality of segmentation results can be measured by applying the user model. Let A_V be the total number of times an assessment c was necessary to perform the transformation from automatic segmentation $\breve{V}_\lambda$ to ground truth V_λ. A_V expresses the amount of labor invested by a user.

The gain $\mathcal{G}$ can now be computed for using an automatic segmentation method compared to the situation in which a video librarian has to segment a video fully manually. Let A'_V be A_V measured for the hypothetical worst case segmentation where $\breve{V}'_\lambda$ consists of one LSU covering the entire movie. Then, gain $\mathcal{G}$ is defined as follows:

$$\mathcal{G}(V) = \frac{A'_V - A_V}{A'_V} \cdot 100\% \tag{4}$$

In a similar way, $\mathcal{G}$ can be computed for a collection of videos by summing all individual measurements for A_V and A'_V.

$\mathcal{G}$ is a powerful criterion. It allows comparison of methods based on one single value and gives a direct measure of economic impact.

D. LSU Segmentation on Incorrect Shot Boundary Segmentation

In this section, we evaluate the necessity of the requirement made in most LSU segmentation methods that ground truth shot boundaries are known. Automatic shot boundary segmentation does not reach 100% correctness [1]. Results should be either adjusted manually before performing LSU segmentation, or the errors in the results should be known not to affect the LSU segmentation significantly. To verify this, we have manually corrected results from an automatic adaptive color histogram based shot segmentation method. Results show 37% false positives and 10% false negatives on average. The results are comparable with the results described in [1] and can be considered state of the art. For more details see [21]. Even in the best case, the adjustments are very labor-intensive. Therefore, the option of manual correction is not viable, unless perfect shot segmentation is required for other applications as well. Hence, evaluation of the effects of incorrect shot segmentation results on LSU segmentation is necessary.

To avoid confusion, we first introduce notation for the different types of segmentation results involved. The ground truth segmentations V_λ for LSUs and V_σ for shots have been described before. The results of automatic LSU segmentation based on V_σ are represented by $\breve{V}_\lambda$. Let the results of automatic shot segmentation be $\hat{V}_\sigma$. The results of automatic LSU segmentation based on automatic shot segmentation $\hat{V}_\sigma$ are then denoted by $\hat{V}_\lambda$. The question is whether $\hat{V}_\lambda$ is sufficiently similar to $\breve{V}_\lambda$, or more general whether the distances from either segmentation to ground truth V_λ are comparable in magnitude. If so, the complete process from raw video data to LSU segmentation can be automated.

For determining the distance between $\hat{V}_\lambda$ and V_λ it is necessary that the underlying shot boundaries correspond. This is non trivial, since an LSU boundary could be detected for a shot boundary in $\hat{V}_\sigma$ that is not present in V_σ. To make comparison of the results possible, we adjust $\hat{V}_\lambda$ such that the following requirement is fulfilled: each boundary in automatic LSU segmentation $\hat{V}_\lambda$ corresponds to the closest ground truth shot boundary in V_σ.

Given the requirements for LSU boundary correspondence, the impact of errors in shot segmentation on LSU segmentation can be evaluated.

TABLE II
OVERVIEW OF VIDEO COLLECTIONS I AND II

Name video	length (minutes)	#shots	#LSUs
Collection I			
A View To A Kill	125	2171	68
Rain Man	108	1166	80
Witness	108	1049	63
Life Of Brian	90	883	37
Fawlty Towers 1-1	30	299	8
Fawlty Towers 2-4	32	420	16
Seinfeld 2-9	22	316	9
Seinfeld 8-20	22	310	25
Simpsons 11-18	22	352	26
Friends 4-17	21	363	16
Friends 4-18	21	328	14
Collection II			
Gladiator	143	3237	91
Forrest Gump	136	1430	154
Conspiracy Theory	127	1974	62
Under Siege II	96	2561	132
Airplane!	84	952	112
Friends 4-19	21	406	13
Overall *I+II*	1208	18217	926

IV. EXPERIMENTS

A. Setup

We use the following implementations for the 4 types of segmentation methods: [2] for overlapping links, [11] (Sections III and IV) for continuous video coherence, [4] for time constrained clustering, and [6] for time adaptive grouping. These methods are most often referred to in literature, and can be seen as first adapters of the particular type. The parameter values suggested in the references were used.

We defined LSU ground truths[1] for 17 popular movies and TV series, in total 20 hours and 926 scenes. The video collection is split into Collections I and II of ten hours each. For Collection I, shot boundaries were manually corrected from automatic results. Collection I is used to evaluate the impact of shot segmentation on LSU segmentation. For Collection II, only automatic shot segmentation results are available. Characteristics of the videos are given in Table II.

B. Features

LSU segmentation depends on computation of visual shot similarity, which in turn is based on visual features. Although implementation details differ, there is consensus in literature on the use of color histograms. In the context of this paper we focus

[1]Tools and ground truths are available via http://www.science.uva.nl/~vendrig/evaluation/

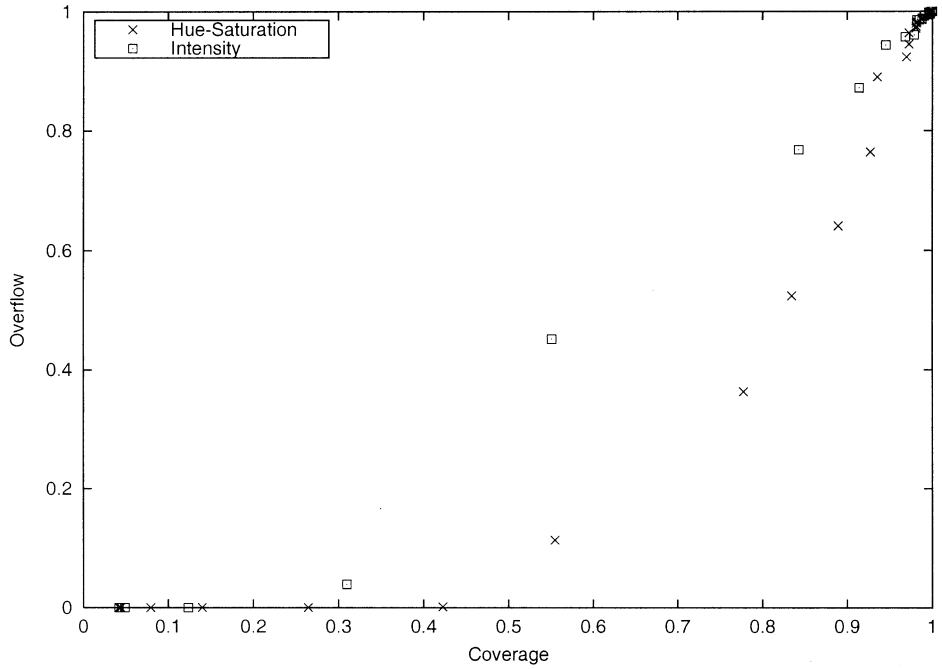

Fig. 3. Coverage $\mathcal{C}$ and overflow $\mathcal{O}$ plotted for two features, measured for 40 thresholds τ^v, median values for ten hours of video.

on evaluating segmentation methods rather than evaluating the quality of features. Therefore, we restrict to evaluation of the Hue–Saturation–Intensity color space. We use two-dimensional Hue–Saturation histograms [9], [6] for the chromatic part of the color space, and Intensity histograms for the achromatic part. The similarity function used is the intersection distance between histograms [22].

C. Results

The coverage $\mathcal{C}$ and overflow $\mathcal{O}$ are plotted against one another for various thresholds in Fig. 3. Apart from outlier video "Fawlty Towers," for all videos a similar trend is visible. "Fawlty Towers" is atypical in the sense that LSUs are long and take place in several settings, while the same settings occur in most LSUs. Since the vast majority of videos exhibit similar results, in Fig. 3 the median values for the video collection are presented for each threshold. The Hue–Saturation histogram feature results in the same coverage as the Intensity histogram feature, but for significantly lower overflow. Therefore, the Hue–Saturation histogram feature is used for further LSU segmentation experiments.

Table III shows the outcome of the evaluation of the segmentation results against the ground truth for Collections I and II. A detailed example of the various results for a small movie segment is given in Fig. 4. Collection I's overall results in Table III show that the performance of methods does not decrease because of incorrect shot segmentation. Overlapping links, and to a lesser extent time constrained clustering, is affected by false negatives (undersegmentation) in $\hat{V}_\sigma$. Shot undersegmentation makes it harder to use visual similarity for shot comparison, as the shot's visual content is diverse. Methods using a binary time window, viz., overlapping links and time constrained clustering, are affected by worse performance of the visual similarity function especially. The error cannot be compensated by other shots with similar visual content if those shots are too far away temporally. False positives (oversegmentation), on the other hand, cause performance to increase, particularly in the case of overlapping links. Oversegmented shots usually have similar visual content. Hence they both are assigned to the same LSU. They do not influence performance negatively. In addition, oversegmentation results in a more precise comparison of shot content, comparable to the shot-lets introduced in [12]. Then the LSU segmentation is more precise as well.

In Table III, the results for Collection II are given as well. Again, the results indicate that there is no necessity for the assumption of a perfect shot boundary segmentation for successful LSU boundary segmentation.

V. Conclusions

LSU segmentation methods are characterized by two dimensions, viz., temporal distance and shot comparison, resulting in four classes. We have defined evaluation criteria for features and segmentation results of the four classes from the perspective of video librarians. For visual features, the evaluation criteria help users in finding thresholds suited for the segmentation process. They also yield insight in the inherent complexity of the segmentation problem. For evaluation of automatic segmentation results, a method is introduced measuring the effort a video librarian should make to convert found segmentation results into a ground truth segmentation.

Given the inherent complexity of segmentation by visual similarity, results are quite good for all methods. Using the gain criterion instead of the traditional counting of under segmentation and over segmentation errors, gives more insight in the economic impact of the errors. Detailed experimental results [21] show that time constrained clustering causes the lowest amount of segmentation errors in total. However, correcting those errors

TABLE III
GAIN $\mathcal{G}$ FOR FOUR LSU SEGMENTATION METHODS PERFORMED ON COLLECTIONS I AND II, BASED ON GROUND TRUTH SHOT SEGMENTATION V_σ, AND AUTOMATIC SHOT SEGMENTATION $\hat{V}_\sigma$. FOR EACH VIDEO, THE GAINS OF THE BEST PERFORMING METHOD IS SHADED

	Segmentation on V_σ				Segmentation on $\hat{V}_\sigma$			
Name video	Overlapping links	Time constrained clustering	Time adaptive grouping	Continuous video coherence	Overlapping links	Time constrained clustering	Time adaptive grouping	Continuous video coherence
Collection I								
A View To A Kill	46%	49%	75%	67%	87%	39%	79%	68%
Rain Man	89%	45%	68%	62%	71%	58%	78%	67%
Witness	53%	65%	80%	69%	87%	76%	83%	84%
Life Of Brian	91%	55%	64%	62%	79%	51%	86%	79%
Fawlty Towers 1-1	75%	77%	75%	88%	71%	76%	88%	85%
Fawlty Towers 2-4	89%	47%	8%	62%	85%	58%	10%	63%
Seinfeld 2-9	93%	67%	69%	72%	80%	64%	61%	72%
Seinfeld 8-20	32%	59%	61%	75%	76%	60%	68%	67%
Simpsons 11-18	16%	56%	75%	78%	67%	53%	83%	85%
Friends 4-17	35%	42%	85%	70%	31%	19%	81%	64%
Friends 4-18	26%	54%	69%	79%	79%	26%	70%	91%
Overall I	61%	54%	69%	68%	78%	52%	75%	74%
Collection II								
Gladiator	-	-	-	-	61%	36%	65%	71%
Forrest Gump	-	-	-	-	95%	47%	55%	44%
Conspiracy Theory	-	-	-	-	58%	46%	75%	68%
Under Siege II	-	-	-	-	93%	17%	64%	59%
Airplane!	-	-	-	-	69%	29%	40%	44%
Friends 4-19	-	-	-	-	37%	44%	49%	58%
Overall II	-	-	-	-	73%	34%	62%	61%

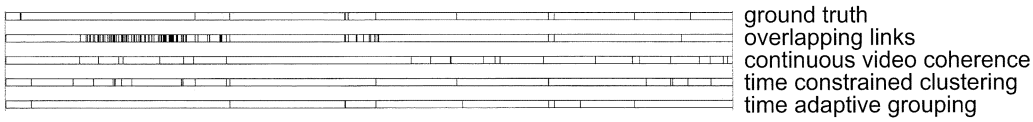

Fig. 4. Example of ground truth and segmentation results for the first 20 minutes of "A View to a Kill" in time lines. Each vertical line represents an LSU boundary.

costs more effort than for other methods due to the magnitude of the errors. The gain criterion is a powerful user-centered measurement of LSU segmentation results.

The overlapping links method's results are unpredictable. Results could be corrected by adapting thresholds for specific movies, but this solution is undesirable as it results in loss of general applicability. Time adaptive grouping shows both good and consistent results. It is the best method to segment a collection of videos.

The use of automatic shot segmentation does not result in significantly worse performance of the LSU segmentation methods as shown in Table III. Hence, the labor-intensive creation of shot segmentation ground truths before performing LSU segmentation is not necessary.

In general, current automatic segmentation methods based on visual features show sound results. Improvement of automatic segmentation is not only sought in development of new visual features, but also in extension with features from other modalities, viz., audio and text. Systematic user centered evaluation should be applied to such multi-modal approaches as well and show to what extent they result in an increase of gain $\mathcal{G}$.

ACKNOWLEDGMENT

The authors would like to thank the development team of the Horus video processing library and J. Baan of TNO-TPD for his shot segmentation tool.

REFERENCES

[1] R. Brunelli, O. Mich, and C. M. Modena, "A survey on the automatic indexing of video data," *J. Vis. Commun. Image Represent.*, vol. 10, no. 2, pp. 78–112, 1999.
[2] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video retrieval systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 580–588, June 1999.
[3] R. M. Bolle, B.-L. Yeo, and M. Yeung, "Video query: Research directions," *IBM J. Res. Develop.*, vol. 42, no. 2, pp. 233–252, 1998.
[4] M. Yeung, B.-L. Yeo, and B. Liu, "Segmentation of video by clustering and graph analysis," *Comput. Vis. Image Understand.*, vol. 71, no. 1, pp. 94–109, 1998.
[5] J.M. Boggs and D. W. Petrie, *The art of watching films*, 5th ed. Mountain View, CA: Mayfield, 2000.
[6] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for videos," *Multimedia Syst., Special Section on Video Libraries*, vol. 7, no. 5, pp. 359–368, 1999.
[7] Y.-M. Kwon, C.-J. Song, and I.-J. Kim, "A new approach for high level video structuring," in *IEEE Int. Conf. Multimedia and Expo*, vol. 2, 2000, pp. 773–776.

[8] S.-Y. Lee, S.-T. Lee, and D.-Y. Chen, *Automatic Video Summary and Description*. Berlin, Germany: Springer-Verlag, 2000, vol. 1929, Lecture Notes in Computer Science, pp. 37–48.
[9] P. Aigrain, P. Joly, and V. Longueville, *Medium Knowledge-Based Macro-Segmentation of Video Into Sequences*. Menlo Park, CA: AAAI, 1997, ch. 8, pp. 159–173.
[10] P. Chiu, Girgensohn, W. Polak, E. Rieffel, and L. Wilcox, "A genetic algorithm for video segmentation and summarization," in *IEEE Int. Conf. Multimedia and Expo*, vol. 3, 2000, pp. 1329–1332.
[11] J. R. Kender and B. L. Yeo, "Video scene segmentation via continuous video coherence," in *CVPR'98*, Santa Barbara, CA, June 1998.
[12] H. Sundaram and S.-F. Chang, "Determining computable scenes in films and their structures using audio visual memory models," in *Proc. 8th ACM Multimedia Conf.*, Los Angeles, CA, 2000.
[13] T. Lin and H.-J. Zhang, "Automatic video scene extraction by shot grouping," in *Proc. ICPR'00*, Barcelona, Spain, 2000.
[14] R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Scene determination based on video and audio features," in *Proc. 6th IEEE Int. Conf. on Multimedia Systems*, vol. 1, 1999, pp. 685–690.
[15] E. Sahouria and A. Zakhor, "Content analysis of video using principal components," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1290–1298, Dec. 1999.
[16] E. Veneau, R. Ronfard, and P. Bouthemy, "From video shot clustering to sequence segmentation," in *Proc. ICPR'00*, vol. 4, Barcelona, Spain, 2000, pp. 254–257.
[17] H. J. Zhang, S. Y. Tan, S. W. Smoliar, and G. Yihong, "Automatic parsing and indexing of news video," *Multimedia Syst.*, vol. 2, no. 6, pp. 256–266, 1995.
[18] J. M. Gauch, S. Gauch, S. Bouix, and X. Zhu, "Real time video scene detection and classification," *Inform. Process. Manage.*, vol. 35, no. 3, pp. 381–400, 1999.
[19] D. Bordwell and K. Thompson, *Film Art: An Introduction*, 5th ed. New York: McGraw-Hill, 1997.
[20] D. A. Adjeroh, I. King, and M. C. Lee, "A distance measure for video sequences," *Comput. Vis. Image Understand.*, vol. 75, no. 1, pp. 25–45, 1999.
[21] J. Vendrig and M. Worring, "Evaluation measurement for logical story unit segmentation in video sequences," Intelligent Sensory Inform. Syst., Univ. Amsterdam, Dept. Comput. Sci., Amsterdam, The Netherlands, 2001–03, 2001.
[22] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.

Jeroen Vendrig (M'00) received the M.Sc. degree in business information systems from the University of Amsterdam, The Netherlands, in 1997. In 2002, he received the Ph.D. degree for his thesis on the "Interactive Exploration of Multi Media Content" project at the University of Amsterdam.

Currently, he is working for MediaMill, a University of Amsterdam spinoff in conjunction with TNO-TPD. MediaMill focuses on development of multimedia indexing tools. His research interests include interactive video segmentation and video retrieval.

Marcel Worring receievd the M.Sc. degree (honors) from the Free University Amsterdam, The Netherlands, in 1988 and the Ph.D. degree from the University of Amsterdam in 1993.

He is an Assistant Professor of computer science at the University of Amsterdam. Main topic of current research is the automatic structuring of the content of multimedia documents to allow for content-based access, exploration, and presentation. In this context, he is leading a large project in which experimentation platforms for large-scale multimedia information analysis (MIA) are being developed. This project is conducted in close relation with industry. He is co-founder of MediaMill, Amsterdam.