IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 6, pp. 904-914, June 2012.

Differential Edit Distance: A metric for scene segmentation evaluation

Panagiotis Sidiropoulos, Vasileios Mezaris, *Member, IEEE*, Ioannis Kompatsiaris, *Senior Member, IEEE*, and Josef Kittler *Member, IEEE*

***Abstract*—In this work a novel approach to evaluating video temporal decomposition algorithms is presented. The evaluation measures typically used to this end are non-linear combinations of Precision-Recall or Coverage-Overflow, which are not metrics and additionally possess undesirable properties, such as non-symmetricity. To alleviate these drawbacks we introduce a novel uni-dimensional measure that is proven to be metric and satisfies a number of qualitative prerequisites that previous measures do not. This measure is named Differential Edit Distance (DED), since it can be seen as a variation of the well-known edit distance. After defining DED, we further introduce an algorithm that computes it in less than cubic time. Finally, DED is extensively compared with state of the art measures, namely the harmonic means (F-Score) of Precision-Recall and Coverage-Overflow. The experiments include comparisons of qualitative properties, the time required for optimizing the parameters of scene segmentation algorithms with the help of these measures, and a user study gauging the agreement of these measures with the users' assessment of the segmentation results. The results confirm that the proposed measure is a uni-dimensional metric that is effective in evaluating scene segmentation techniques and in helping to optimize their parameters.**

I. Introduction

Video decomposition into elementary temporal units is an essential preprocessing task for a wide range of video manipulation applications, such as video indexing, non-linear browsing, classification etc. The video decomposition techniques focus either on shot or scene segmentation, according to the structural or semantic criteria employed.

Shots are defined as sequences of images taken without interruption by a single camera [1]. On the other hand, scenes are longer temporal segments that are usually defined as Logical Story Units (LSU): a series of temporally contiguous shots characterized by overlapping links that connect shots with similar content [2]. The video scene should not be confused with the meaning of the term "scene" in the context of still image processing and interpretation, which relates to the physical environment information that is captured by the image.

Automatic video segmentation to shots and scenes is associated with different degrees of difficulty. State-of-the-art shot segmentation techniques have been shown to reach good performance on a variety of datasets in experiments such as the annual TRECVID benchmarking exercise, particularly when it comes to detecting abrupt shot transitions (cuts) [3]. On the other hand, scene segmentation is still an open research problem. Among the shortcomings of the relevant scene segmentation literature is the lack of an efficient scene segmentation evaluation measure.

Automatic scene segmentation techniques generate a list of scene boundaries that identify the time-points dividing the video stream into different scenes. In order to estimate their performance, the resulting scene boundary list is contrasted with a manually generated one (ground truth). The similarity of the two scene boundary lists is measured either in terms of Precision-Recall [4] of Coverage-Overflow [5]. In [6], editing strategies common to film industry are exploited to extract the scene boundaries and the results are evaluated by using Precision-Recall (and a linear combination of them), as well as the required computation time. In [7] a visual bag-of-words approach is proposed for decomposing the video into scenes, which for the purpose of evaluation are compared to the ground-truth using the Coverage and Overflow measures. The authors of [8] present a graph-based scene segmentation approach, which uses normalized cuts; evaluation is conducted with the help of the Precision-Recall measures. [9] proposes a probabilistic technique that aims to maximize the Precision-Recall values of the estimated scene boundaries, by training a number of independent descriptors based on various modalities, with Precision-Recall again being used for its evaluation. Similarly, the authors of [10] train a SVM, which takes as input descriptor values from different modalities, to maximize the Precision-Recall measures. Finally, a multi-modal probabilistic technique that uses both high-level and low-level audio-visual features (including visual concepts and audio events, automatically detected with the use of a plurality of machine-learning-based concept and event detectors) is proposed in [11]. Its evaluation is carried out using Coverage and Overflow.

For most of the aforementioned methods, as well as other techniques of the relevant literature, when a straightforward,

P. Sidiropoulos is with the Informatics and Telematics Institute / Centre for Research and Technology Hellas, 6th Km Charilaou-Thermi Road, P.O.BOX 60361, Thermi 57001, Greece, and with the Center for Vision, Speech and Signal Processing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey GU2 5XH, UK, E-mail: psid@iti.gr, P.Sidiropoulos@surrey.ac.uk

V. Mezaris, I. Kompatsiaris are with the Informatics and Telematics Institute / Centre for Research and Technology Hellas, 6th Km Charilaou-Thermi Road, P.O.BOX 60361, Thermi 57001, Greece, E-mail: bmezaris, ikom@iti.gr.

J. Kittler is with the Center for Vision, Speech and Signal Processing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey GU2 5XH, UK, E-mail: J.Kittler@surrey.ac.uk.

This work was supported by the European Commission under contract FP7-248984 GLOCAL. A partial support from EPSRC Project EP/F069421/1 is also gratefully acknowledged.

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6111460

uni-dimensional comparison is required, for example to optimize the value of a system parameter, the harmonic mean [1](F-Score) of either the one or the other of these two pairs of measures is typically estimated. However, this evaluation approach suffers from a number of evaluation flaws, which are partially induced by the fact that the generated problem space (i.e., the evaluation space) is not a metric space.

In this work we present a novel uni-dimensional measure for scene segmentation evaluation, along with an implementation of it that features lower-than-cubic complexity. We prove that this measure is a metric and we compare it with the two aforementioned harmonic means commonly used in the literature, to demonstrate its desirable properties and its increased agreement with the users' evaluation of segmentation results. We also experimentally show that the tuning of scene segmentation system parameters using the new measure requires less time, since it allows for a more sparse sampling of the parameter space.

The rest of the paper is organized as follows: The concept of scene segmentation as a label assignment problem, which is a prerequisite for the development of the metric proposed in this work, is discussed in Section II. The Differential Edit Distance metric and its estimation algorithm are presented and discussed in Section III, followed in Section IV by experimental evaluation and comparison with two other uni-dimensional evaluation measures. Finally, conclusions are drawn in Section V.

II. Scene Segmentation as a Label Assignment Problem

Mathematically speaking, a video sequence V can be seen as a well-ordered set of structural elements such as frames, shots, scenes. That is, considering only one of the aforementioned possible types of elements at each time, their set has a binary relation $\Re$ that is total (for all $x_i, x_j \in V$, $x_i \Re x_j$ or $x_j \Re x_i$), antisymmetric (if $x_i \Re x_j$ and $x_j \Re x_i$, then $x_i = x_j$) and transitive (if $x_i \Re x_j$ and $x_j \Re x_k$, then $x_i \Re x_k$). This binary relation is the temporal position of the video's structural elements.

Video temporal decomposition techniques generate a partition of video sequence V into convex sub-sets v_i, since the resulting temporal segments (regardless of whether they are shots or scenes) by definition satisfy the following principles:

- $\bigcup v_i = V$
- $v_i \bigcap v_j = \emptyset$, $\forall i \neq j$
- $\forall v_i$ if $x_1, x_2 \in v_i$ then all x, $x_1 \leq x \leq x_2$ also belong to v_i

The first two principles signify that each and every video element is assigned into one of non-overlapping sub-sets (considering, of course, only the appropriate types of elements for each task). For example there are no frames that do not belong to some shot, scene etc. Finally, the third one is associated with the sub-set convexity, since it postulates that if two elements belong to the same sub-set then all elements that lie between them also belong to it.

When shot segmentation is conducted, the video elements considered are the frames of the video. On the other hand, when scene segmentation is conducted, the video elements considered are usually the video shots. This reflects a common assumption behind almost all scene segmentation techniques in the literature, namely that each shot belongs to exactly one scene [11], [12], [13], [14], [15]. Under this assumption scene segmentation is typically performed through a two-step temporal decomposition process: first the video frames are used to partition the video sequence into shots, and then the shots are further grouped to form scenes. In the second step of this approach, each shot is assigned to an appropriate scene. We can assume that this is performed through a labeling process: each shot receives a label that identifies the scene that it belongs to, so that:

- If two shots belong to the same scene, they are assigned the same label.
- If two shots belong to different scenes, they are assigned different labels.

For example, a video sequence that includes 5 shots may be labeled "a, a, b, b, c", "$1, 1, 1, 1, 1$", etc. On the other hand, the label sequences "a, b, b, c" and "a, a, b, b, a" do not represent possible decompositions of this video into scenes: in the first case one shot is not assigned to any scene, while in the second case the decomposition is not a convex one.

So, scene segmentation can be generally viewed as a label assignment problem, where one is interested in estimating a label sequence that corresponds to the grouping of the video's shots into scenes. This scene segmentation approach is in line with the point of view of an expert user, who is charged with the generation of a manual segmentation of a video stream (e.g. a video librarian [5]). Such a user would assign labels into scenes and would discriminate one scene from another by moving from shot to shot while changing the assigned description label only when the scene changes.

III. Differential Edit Distance Metric

A. Differential Edit Distance

In any objective scene segmentation evaluation setup, the ground-truth scene segmentation and the experimentally estimated scene segmentation results provide two different partitions of the well-ordered set of shots. The similarity of these partitions may be used as a measure of accuracy of the experimentally estimated scene segmentation. We propose to express this similarity through a minimum distance approach that resembles the Earth Movers' Distance; the latter was recently used, among others, for visual event recognition and near-duplicate video detection [16], [17]. More specifically, we define the distance between two partitions of a well-ordered set as the minimum number of set elements that need to move to another sub-set in order to transform the one partition into the other. Using the scene segmentation terminology, the distance between two scene segmentation partitions is the minimum number of shots that need to change scene label in order to transform the experimentally estimated partition into the ground truth one.

[1] It is reminded that the harmonic mean F_{Q_1,Q_2} of two quantities Q_1 and Q_2 is $F_{Q_1,Q_2} = \frac{2Q_1Q_2}{Q_1+Q_2}$.

It can be proven that this measure is also a metric. Indeed, if $d(V_1, V_2)$ denotes the distance between two partitions V_1 and V_2 then it is obvious that $d(V_1, V_1) = 0$ and $d(V_1, V_2) = d(V_2, V_1)$. Furthermore, let us suppose that X_{ij} is the set of elements giving rise to $d(V_i, V_j)$ (set of elements that need to change sub-set). Then, if X_{12} and X_{23} are two such sets, by changing a sub-set of all elements that belong to $X_{12} \bigcup X_{23}$ the partition V_1 can be transformed into partition V_3. Since the distance $d(V_1, V_3)$ is the minimum number of elements that need to change subset, $d(V_1, V_3) \leq |X_{12} \bigcup X_{23}|$ (where $|.|$ denotes the cardinality of a set). Moreover, each element that belongs to $X_{12} \bigcup X_{23}$ must change subset in order to transform either V_1 into V_2 or V_2 into V_3 or both. Consequently:

$$d(V_1, V_3) \leq |X_{12} \bigcup X_{23}| \leq d(V_1, V_2) + d(V_2, V_3) \qquad (1)$$

We name this metric Differential Edit Distance (DED) due to the fact that when video partitioning is modeled as a label assignment problem, then this distance expresses the minimum number of labels that need to change in order to transform the first label sequence into another that achieves an identical partitioning with the second. It can be seen from this definition that DED resembles the well-known edit distance [18]. The edit distance differs from DED in that it additionally requires the identical partitioning to be expressed with identical labels. In order to give a definition that is tailored to label assignment, we first introduce Differential Equivalence:

Definition 1: Two label sequences are differentially equivalent when each pair of elements in the two sequences satisfies the following conditions:

- If the two elements of the pair share the same label in the first sequence (i.e., if $label\ of\ x_i = label\ of\ x_j$ according to the first label sequence), they will also have a common label in the second sequence (i.e., $label\ of\ x_i = label\ of\ x_j$ also according to the second label sequence. The latter common label may of course be different from the one that the two elements shared according to the first label sequence.)
- If they do not share the same label in the first sequence (i.e. if $label\ of\ x_i \neq label\ of\ x_j$ according to the first label sequence), they will also have different labels in the second sequence (i.e., $label\ of\ x_i \neq label\ of\ x_j$ also according to the second label sequence).

For example label strings “a, a, b, b, c, c”, “$1, 1, 2, 2, 3, 3$”, “$2, 2, 1, 1, 3, 3$”, “$B, B, 1, 1, A, A$”, “$+, +, -, -, *, *$” are all differentially equivalent. Differentially equivalent label sequences correspond to identical set partitions.

DED is then defined as the minimum number of label modifications that are required to transform the first label sequence into a sequence that is differentially equivalent to the second one.

As discussed above, DED is a metric measure. It is assumed here that evaluating scene segmentation methods with a metric measure can be advantageous in comparison to using non-metric ones. One of the reasons for this is that when a metric measure is used for guiding an optimization process (as will be examined in section IV-D), it is intuitively expected to result in an error signal of lower bandwidth. Thus, estimation of the measure values at fewer points of the parameter space is sufficient for finding a good solution to the optimization problem. While the validity of this assumption is not guaranteed, the experimental results of section IV-D indicate that the proposed metric measure indeed results in most cases in an error signal of lower bandwidth, in comparison to non-metric measures F_{PR}, F_{CO}. Furthermore, if one needed to process the samples of this error signal in a more elaborate way than what is done in this work, e.g. if one wanted to perform some kind of machine learning or dimensionality reduction involving these samples, the fact that they define a metric space allows for the use of techniques such as SVM, PCA or isometrical embedding [19], [20], [21], which are designed specifically for use in metric spaces.

B. DED Estimation Algorithm

The DED algorithm computes the minimum number of labels that need to change in order to transform one label sequence into another. As will be subsequently demonstrated, this problem can be solved in less than cubic time by modeling it as a job assignment problem. The final resulting algorithm is summarized in Algorithm 1.

Let us suppose that the alphabet (i.e. the set of labels) of the experimentally estimated label sequence and the ground truth one is A_E and A_G respectively and that the number of labels in each alphabet is $|A_E|$ and $|A_G|$. Since DED is symmetric, the experimentally estimated label sequence and the ground truth one can switch places without changing the final DED outcome. Consequently, we can assume that $|A_E|$ is larger than $|A_G|$ without loss of generality.

Each symbol a_i^g, $i \in \{1, 2, ..., |A_G|\}$ of the ground truth label sequence is used to label the shots that belong to a ground truth scene (i.e., label a_i^g is the one assigned to the shots of ground truth scene v_i^g; both labels and scenes are ordered according to the temporal order of the scenes in the video, so that a_1^g is the label of the first scene (v_1^g), a_2^g of the second one, etc.). The shots assigned to a_i^g according to the ground truth label sequence are also assigned to labels $a_j^e, a_{j+1}^e, ..., a_{j+k}^e$ in the experimental label sequence. It is obvious that from all $k+1$ labels $a_j^e, ..., a_{j+k}^e$, at most one can be considered to correspond to ground truth label a_i^g. If this is $a_{j'}^e$, we say that label $a_{j'}^e$ is a match to label a_i^g. Each symbol in the experimental sequence can match at most with one symbol of the ground truth sequence and vice versa (the exact way that this matching is performed is explained in the sequel).

Following the label matching, all shots that belong to a scene labeled a_i^g and whose experimentally assigned label belongs to set $\{a_j^e, ..., a_{j+k}^e\} - \{a_{j'}^e\}$ need to change their label. In case there is no match for label a_i^g, all shots belonging to this ground truth scene need to change their label. Consequently, for all i, if a_i^g is matched with a label belonging to the experimental label set, the number of shots that need to change label is equal or less than the respective number of shots that would need to change label if a_i^g had not been

Algorithm 1 DED Algorithm Summary

```
1: If $B_E$ and $B_G$ are the ordered sets (in ascending order) of experimentally estimated scene boundaries and ground truth scene boundaries respectively, ordered set $B = \{0, B_E \cap B_G, N\}$ is formed, where $N$ is the number of shots in the video. It should be noted that in sets $B_E$, $B_G$, a scene boundary is represented by the index of the last video shot that is part of the first of the two scenes defining the boundary.
2: The video is decomposed into sub-videos $SV_b, b = 1, 2, ...|B| - 1$, where $|B|$ is the cardinality of $B$. Each sub-video starts at shot $B(b) + 1$ and lasts until the end of shot $B(b + 1)$. The way that this decomposition is performed is discussed in section III-C.
3: Initialize: $b = 1$, $N_W = 0$.
4: For the sub-video $SV_b$, a co-occurrence matrix, $CM_b$, is constructed. Each element $CM_b(i, j)$ of the co-occurrence matrix is equal to the number of shots that belong to both ground truth scene $v_i^g$ and experimental scene $v_j^e$.
5: Cost matrix $CC_b$ is computed as $CC_b(i, j) = \hat{CM}_b - CM_b(i, j)$, where $\hat{CM}_b = \max_{i,j}(CM_b(i, j))$.
6: The cost matrix is zero-padded in order to become square.
7: The Hungarian algorithm is used to estimate the element combination that leads to the minimum cumulative cost when choosing only one element of each row and each column of the cost matrix $CC_b$. This combination determines the optimal matching $W_b$ between ground truth and experimentally estimated scenes of the sub-video $SV_b$.
8: The number of shots $N_{W_b} = \sum_{(v_i^g, v_j^e) \in W_b} CM_b(i, j)$ that do not need to change scene label is estimated.
9: $N_W = N_W + N_{W_b}$.
10: If $b = |B| - 1$, $DED = (N - N_W)/N$. Else $b = b + 1$ and the algorithm continues from step $4$.
```

matched with any label from A_E. As a result, in the minimum label modification case, all a_i^g are matched to exactly one label from A_E.

Accordingly, we construct a co-occurrence matrix CM of dimensions $|A_G| \times |A_E|$. Element $CM(i, j)$ contains the number of shots that are assigned the $i - th$ label of alphabet A_G in the ground truth label sequence and the $j - th$ label of alphabet A_E in the experimental label sequence. The value of each element of the co-occurrence matrix is therefore equal to the number of labels that would not require changing if the corresponding symbols a_i^g, a_j^e were considered to match. Consequently, the minimization of the number of transformations is equivalent to the selection of $|A_G|$ matching pairs of symbols maximizing the number of labels that would not need to be changed. This selection is constrained by the fact that each symbol of the one alphabet can be matched at most to one symbol of the other.

Thus, DED estimation leads to the dual problem of job assignment. Let us recall that in the job assignment problem a number of employees need to be assigned to a number of jobs in order to minimize the total cost, with the constrain that each employee can be assigned to no more than one job. The optimal job assignment can be estimated by the Hungarian algorithm [22]. This algorithm takes as input a square matrix with positive elements and estimates with cubic complexity the minimum sum that can be achieved when from each row and each column exactly one element is added. In our case the co-occurrence matrix is transformed into a cost matrix by replacing all values $CM(i, j)$ with $\hat{CM} - CM(i, j)$, where $\hat{CM} = \max_{i,j}(CM(i, j))$ (step 5 of Algorithm 1). Then, the optimal set of symbol matchings is revealed by the element combination that achieves the minimum score according to [22], and is used to estimate the actual DED value from the co-occurrence matrix:

$$DED = \frac{N - N_W}{N} \tag{2}$$

where N is the total number of video shots and N_W is the number of video shots that are assigned labels which are matched correctly.

C. DED Computational Optimization

The job assignment problem solved by the Hungarian algorithm has cubic computational complexity, determined by the minimum number of actual and experimentally estimated scenes. Since the number of scenes is not expected to surpass the order of hundreds, the computational cost is usually not expected to reach extreme levels. However, there may be cases, e.g. when tuning the parameters of a scene segmentation system, that this computational complexity makes the use of DED troublesome. We have found that the DED computational cost can be significantly reduced if the block-diagonal structure of the co-occurrence matrix is exploited.

The co-occurrence matrix structure is induced by “splitting” shot boundaries, i.e. shot boundaries that both in the experimental and the ground truth segmentation are identified as scene boundaries (Fig. 1). It can be proven that all the labels on the left side of a “splitting” boundary do not co-occur with the labels on the right side of it, due to the scene convexity. Consequently, the video stream can be decomposed into sub-videos. This is done by checking the sets of ground truth and experimentally estimated scene boundaries for common boundaries, i.e, we find the scene boundaries that belong to the intersection of these two sets. The latter scene boundaries are used as splitting points for decomposing the video into sub-videos: each such boundary marks the end of a sub-video. The resulting decomposition is illustrated for an example video in Fig. 1.

Consequently, if the scene labels are sorted by their first appearance, the co-occurrence matrix CM takes the following block-diagonal form, where $SV_b, b = 1, 2, ...|B| - 1$ is the $b-$th sub-video, $|B| - 1$ is the total number of sub-videos (see steps 1 and 2 of Algorithm 1 for a definition of $|B|$) and each sub-video boundary is determined by a corresponding “splitting” boundary.

In this case, the optimal job assignment can be estimated by decomposing the co-occurrence matrix into the block-matrices found on its main diagonal, computing the optimal solution for

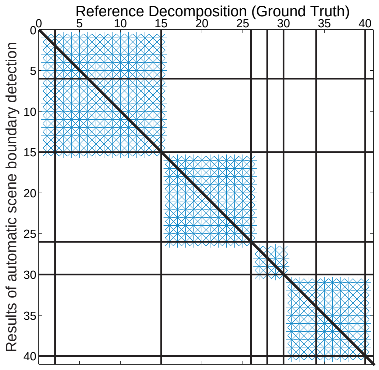
Reference Decomposition (Ground Truth)
0
5
10
15
20
25
30
35
40
Results of automatic scene boundary detection

Fig. 1. An example of a video stream decomposition into sub-videos, using the common scene boundaries of ground truth and experimental segmentation. It should be noted that this figure does not depict a co-occurrence matrix, since its axis indicate shot indices rather than scenes. The vertical and horizontal lines signify the shots that define the ground truth and experimentally estimated scene boundaries respectively. The video is decomposed in points where a vertical and a horizontal line intersect on the main diagonal. Each sub-video is drawn hatched.

$$CM = \begin{bmatrix} SV_1 & 0 & 0 & ... & 0 \\ 0 & SV_2 & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & SV_{|B|-1} \end{bmatrix}$$

each SV_b, $b = \{1, 2, ...|B| - 1\}$ matrix, and summing all the partial solutions.

It should be noted that the technique presented in this section is used for evaluating the segmentation similarity when the cost of a shot re-assignment is assumed identical and equal to 1. However, the same analysis stands if the shot re-assignment cost is determined by specific shot-related criteria, such as the shot duration in frames or seconds. In this case, only the co-occurrence matrix calculation (step 5 of Algorithm 1) needs to be modified in order to represent these costs, counted in e.g. seconds rather than in number of shots.

IV. Comparison of Scene Segmentation Evaluation Measures

In the early scene segmentation literature, the evaluation of segmentation results was either subjective (e.g. in [23] it was left to the reader) or was based on evaluation criteria for shot boundary segmentation. The latter boils down to counting false negatives and false positives (e.g. in [2]) that leads to a Precision-Recall approach. In some recent publications, (e.g. in [11]), instead of Precision-Recall, Coverage and Overflow measures [5] are employed.

However, when the performance is evaluated by two distinct measures, the inherent problem of combining them needs to be addressed. In both Precision-Recall and Coverage-Overflow based approaches, their harmonic mean has been proposed as a uni-dimensional measure combining the two. In the following sub-section, the DED is comparatively evaluated against the harmonic mean of Precision-Recall (F_{PR}) and Coverage-Overflow (F_{CO}).

A. Other Scene Segmentation Evaluation Measures

1) Precision-Recall: Precision and Recall [4] are two widely used performance measures (e.g. see [12], [13], [24]). They require a set of ground truth instances and a set of experimentally estimated instances. For scene segmentation purposes, we have chosen to relate the set of ground truth and experimentally estimated instances with the pairs of shots that belong to the same scene, since each video scene segmentation explicitly determines the shot pairs that belong to the same scene.

It should be noted that in the relevant literature, Precision and Recall are commonly estimated by counting false positives and false negatives in the experimentally retrieved set of scene boundaries, rather than pairs of shots that belong to the same scene. However, this approach can not correctly gauge scene segmentation performance, since the number of errors does not communicate error magnitude [5]. Misidentified scene boundaries represent errors of different magnitude, which are expected to play a different role to the system performance. By defining Precision and Recall with the help of pairs of shots that belong to the same scene, cases such as the above can be handled successfully. However, as will be discussed in the following subsection, even when using such a definition the harmonic mean of these two measures continues to present both theoretic and experimental shortcomings in comparison to the DED.

2) Coverage-Overflow: Vendrig et. al. [5] developed two novel measures that manage to express over-segmentation and under-segmentation rates, referred to as Coverage and Overflow ratio. Coverage (C) measures to what extent frames belonging to the same scene are correctly grouped together, while Overflow (OV) evaluates the quantity of frames that, although not belonging to the same scene, are erroneously grouped together. More specifically, the Coverage and Overflow of a video is the average Coverage and Overflow ratios of its ground truth scenes. In order to estimate the Coverage and Overflow of a ground truth scene v_i^g, the experimentally estimated scenes $v_j^e, v_{j+1}^e, ..., v_{j+k}^e$ that overlap with it are taken into account. Then, if operator $||.||$ denotes the duration of a video segment (counted in shots), the Coverage C equals the maximum overlap divided by the total scene duration:

$$C_i = \frac{max(||v_j^e \cap v_i^g||, ||v_{j+1}^e \cap v_i^g||, ..., ||v_{j+k}^e \cap v_i^g||)}{||v_i^g||} \quad (3)$$

On the other hand, in order to compute the Overflow rate, the total overlap of $v_j^e, v_{j+1}^e, ..., v_{j+k}^e$ with the scenes neighboring to v_i^g (i.e. v_{i+1}^g and v_{i-1}^g) is estimated and is divided by the duration of these scenes:

$$OV_i = \frac{||v_j^e \cap v_{i+1}^g|| + ||v_j^e \cap v_{i-1}^g|| + ...||v_{j+k}^e \cap v_{i-1}^g||}{||v_{i+1}^g|| + ||v_{i-1}^g||} \quad (4)$$

It should be noted that Coverage and Overflow optimal values are 100% and 0% respectively. In order to account for 0 being the optimal Overflow value, in the F-score estimation formula the quantity $1 - OV$ is used instead of OV.

B. Evaluation setting

Assessing an evaluation method, such as the one proposed here, is by no means a straightforward process. In the relevant literature there are neither detailed qualitative explanations nor experimental results that would provide supporting evidence for the superiority of one or the other measure [5]. We have chosen to address this problem by following an evaluation setting that involves both qualitative and experimental comparison. The former is performed by identifying a number of qualitative properties that a good measure is intuitively expected to satisfy and checking whether they are exhibited by the proposed method (and the other methods in the literature), while the latter revolves around examining the processing time that is required for tuning the parameters of a scene segmentation system when one of the aforementioned measures is used for guiding the parameter selection process. A user study involving 6 non-expert users was also conducted.

In order to compare the three measures, we implemented four different scene segmentation techniques, and used them on three datasets. The scene segmentation techniques include the original STG technique [23], an STG variation that employs high-level audio event descriptors instead of low-level visual descriptors, as described in [11], and two multi-modal scene segmentation techniques [25], [26]. The video datasets are a documentary, a movie and a news one. The first is made of 15 documentaries (513 minutes in total) from the collection of the Netherlands Institute for Sound & Vision, also used as part of the TRECVID dataset in 2009. The second one is made of six movies (643 minutes in total). Finally, the news dataset consists of 3 hour-long news videos. These datasets include 3459, 6665 and 1763 automatically detected shots, and 525, 357 and 57 manually identified ground truth scenes, respectively. It should be noted that in the news and movie datasets the ground truth scenes usually include many more shots than in the documentary one.

All experiments reported in the sequel were carried out on a PC with an Intel Core 2 Quad Q9300 CPU and 4GB of RAM.

C. Analysis of qualitative properties of evaluation measures

In this subsection the comparison of DED, F_{CO} and F_{PR} according to certain qualitative properties is conducted. It should be noted that since DED is a dissimilarity measure, while F_{CO} and F_{PR} are similarity measures, $1 - DED$ is employed instead in the comparisons.

1) Symmetry in scene boundary misidentification errors: An example of a misidentification error is demonstrated in Fig. 2. The scene boundary which exists at the end of shot S_1 is misplaced by e shots, being detected either at the end of shot $S_1 - e$ or at the end of shot $S_1 + e$. It is reasonable to expect that a good evaluation measure does not discriminate between these two cases, i.e. that it generates identical results without taking into account whether the estimated scene boundary is found before or after the actual one. As a matter of fact, there is no rationale that could support any differentiation of the two cases.

It can be proven that if a scene boundary that exists at the end of shot S_1 is erroneously detected at the end of shot $S_1 - e$, the harmonic mean of Coverage and Overflow, $F_{CO}(v_1, v_2, e)$, is:

$$F_{CO} = \frac{2}{||v_1|| + ||v_2||} \cdot \frac{||v_2||(||v_1|| + ||v_2|| - e)(||v_1|| - e)}{(||v_1||^2 + 2||v_1||||v_2|| - e(||v_1|| + ||v_2||))} \quad (5)$$

where $||v_1||$ and $||v_2||$ is the duration, counted in shots, of the scene to the left and to the right of the scene boundary, respectively. Based on the above equation, $F_{CO}(v_2, v_1, e)$ gives the harmonic mean if the scene boundary is detected at the end of shot $S_1 + e$ instead. Since this formula is not symmetric, F_{CO} generates different scores for equivalent errors, e.g. for the case of $||v_1|| = 30$, $||v_2|| = 70$ and $e = 3$, $F_{CO}(v_1, v_2, e) = 0.7323$ and $F_{CO}(v_2, v_1, e) = 0.4388$.

Symmetry in scene boundary misidentification errors is also not satisfied by measure F_{PR}. When a scene boundary that exists at the end of shot S_1 is erroneously detected at the end of shot $S_1 - e$, the harmonic mean of Precision and Recall, F_{PR}, is:

$$F_{PR}(v_1, v_2, e) = \frac{Q(v_1, v_2, e)}{Q(v_1, v_2, e) + e(||v_1|| + ||v_2|| - 1)} \quad (6)$$

where $Q(v_1, v_2, e) = ||v_1||^2 + ||v_2||^2 + e^2 - (2e + 1)||v_1|| - ||v_2|| + e$. In the above equation $F_{PR}(v_2, v_1, e)$ gives the harmonic mean if the scene boundary is detected at the end of shot $S_1 + e$. Equation (6) is not symmetric, because quantity Q is not symmetric. Consequently, the F_{PR} measure also generates different distance scores for equivalent errors.

On the other hand, DED by definition does not discriminate between these types of errors and produces in both cases a similarity value proportional to the error magnitude:

$$DED(v_1, v_2, e) = DED(v_2, v_1, e) = \frac{e}{(||v_1|| + ||v_2||)} \quad (7)$$

In order to quantify the expected asymmetry, for all videos belonging to the 3 datasets that we use in this work, pairs of synthetic segmentations were constructed by introducing symmetric misplacements of each ground truth scene boundary. Specifically, starting from the ground truth segmentation and considering one scene boundary at a time, this boundary was misplaced by e and $-e$ shots, respectively, where e was selected randomly from the integer values that are smaller than the minimum distance of that particular scene boundary from its two adjacent scene boundaries (so that the introduced misplacement would not lead to a violation of the scene convexity restriction). A single value of e was of course used for each pair of scene boundary misplacements, to ensure their symmetry. Then, DED, F_{PR} and F_{CO} values were estimated (always in the range 0 to 100%) by comparing each synthetic segmentation with the ground truth one, and subsequently the DED, F_{PR} and F_{CO} differences were calculated for each pair of synthetic segmentations that present symmetric errors. The mean and standard deviation of these differences, post-processed so as to simulate the case where 25% of the true scene boundaries of each video are misplaced in this way, are reported separately for each video dataset in Table I.

Dataset	DED Diff. ($\mu \pm \sigma$)	F_{PR} Diff. ($\mu \pm \sigma$)	F_{CO} Diff. ($\mu \pm \sigma$)
Documentary	0 ± 0	0.83% ± 1.62%	6.81% ± 7.92%
Movie	0 ± 0	0.61% ± 1.17%	5.17% ± 5.4%
News	0 ± 0	0.34% ± 0.47%	2.78% ± 2.21%

TABLE I
EXPERIMENTALLY ESTIMATED MEASURE DIFFERENCES FOR SEGMENTATION PAIRS THAT PRESENT SYMMETRIC SCENE BOUNDARY MISIDENTIFICATION ERRORS.

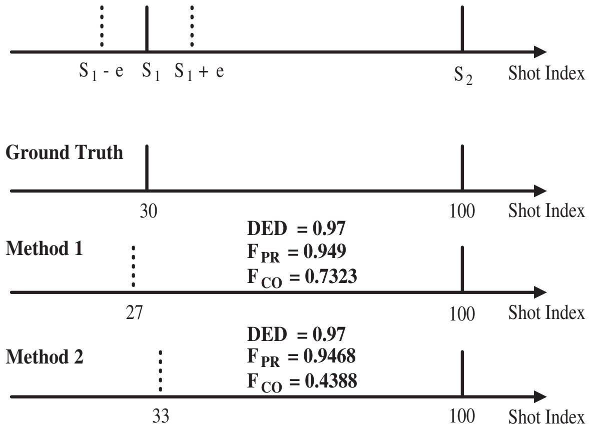

Fig. 2. An example of a misidentification error evaluation with F_{CO}, F_{PR} and DED. Vertical bars denote scene boundaries; the dotted vertical bars represent erroneously detected ones. Quantities S_1 and S_2 denote the shot indices of the last shot of the first and second scene, respectively. While both scene segmentation methods 1 and 2 misidentify the scene boundary by 3 shots, only DED generates symmetric results.

2) Symmetry of errors located at the beginning and the end of a scene: This property is similar to the one discussed above. A scene segmentation technique should not be evaluated differently if it "crops" the beginning or the end of a specific scene. An example of this is shown in Fig. 3.

In order to quantify the expected asymmetry magnitude between errors taking place at the beginning and the end of a scene, an experimental strategy analogous to the previous subsection was followed, where symmetric errors were similarly introduced to each pair of adjacent scene boundaries. The mean and standard deviation of the resulting DED, F_{PR} and F_{CO} differences, as in the previous experiment, are reported separately for each video dataset in Table II.

Dataset	DED Diff. ($\mu \pm \sigma$)	F_{PR} Diff. ($\mu \pm \sigma$)	F_{CO} Diff. ($\mu \pm \sigma$)
Documentary	0 ± 0	7.17% ± 8.39%	18.8% ± 14.36%
Movie	0 ± 0	9.2% ± 12.87%	18.97% ± 16.33%
News	0 ± 0	2.94% ± 3.59%	9.92% ± 11.25%

TABLE II
EXPERIMENTALLY ESTIMATED MEASURE DIFFERENCES FOR SEGMENTATION PAIRS THAT PRESENT SYMMETRIC ERRORS AT THE BEGINNING AND AT THE END OF A SCENE.

As demonstrated by the results of Table II and also the example of Fig. 3, only DED satisfies this property. Employing F_{CO} or F_{PR} leads to different (non-symmetric) performance estimates, induced by the different lengths of the adjacent scenes.

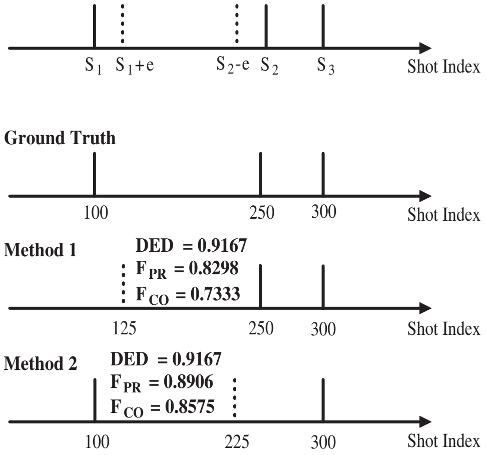

Fig. 3. An example of a misidentification error evaluation with F_{CO}, F_{PR} and DED. Vertical bars denote scene boundaries; the dotted vertical bars represent erroneously detected ones. Quantities S_1, S_2 and S_3 denote the shot indices of the last shot of the first, second and third scene, respectively. Method 1 misplaced the beginning of the second scene by 25 shots, while method 2 misplaced the same scene's end by 25 shots. The two methods are evaluated differently by F_{CO} and F_{PR}.

3) Satisfaction of metric property: In section III-A it was proven that the DED measure is a metric. On the contrary, F_{CO} is not a metric, since it is not symmetric. For example, if a video stream consists of 100 shots and one scene and the experimental segmentation identifies two equally-long scenes, then $F_{CO} = 0.667$. In the opposite case, i.e. when a video stream includes two scenes of 50 shots each and a scene segmentation technique retrieves only one scene, then $F_{CO} = 0$. So, generally

$$F_{CO}(V_1, V_2) \neq F_{CO}(V_2, V_1) \qquad (8)$$

where V_1 and V_2 are two segmentations of the same video stream.

On the other hand, measure F_{PR} satisfies the symmetry property. This is proven by considering the definition of Recall and Precision as the ratio of the intersection of the sets of ground truth and experimental shot pairs belonging to the same scene over the ground truth V_G and the experimental set V_E, respectively:

$$R(V_G, V_E) = \frac{|V_G \cap V_E|}{|V_G|}, P(V_G, V_E) = \frac{|V_G \cap V_E|}{|V_E|} \qquad (9)$$

F_{PR} is defined as the harmonic mean of Recall R and Precision P:

$$F_{PR}(V_G, V_E) = \frac{2}{\frac{1}{R} + \frac{1}{P}} = \frac{2|V_E \cap V_G|}{|V_E \cup V_G| + |V_E \cap V_G|} \qquad (10)$$

$F_{PR}(V_E, V_G)$ estimates the similarity of the two segmentations. The corresponding distance $D_{PR}(V_E, V_G)$ is given by the following equation:

$$D_{PR} = 1 - F_{PR} = \frac{|V_E \cup V_G| - |V_E \cap V_G|}{|V_E \cup V_G| + |V_E \cap V_G|} \qquad (11)$$

It is straightforwardly understood that $D_{PR}(V_G, V_E) = D_{PR}(V_E, V_G)$ and as a result the measure exhibits the symmetry property. However, the distance D_{PR} does not generally satisfy the triangular inequality. For example, let us suppose that a video stream consists of four shots, and three different segmentations V_1, V_2 and V_3 have been defined for it:

$$V_1 = \{1, 2\}, \{3\}, \{4\}$$

$$V_2 = \{1, 2\}, \{3, 4\}$$

$$V_3 = \{1\}, \{2\}, \{3, 4\}$$

In the above equations, the brackets denote scene boundaries. For segmentations V_1 and V_3, the intersection of shot pairs that belong to the same scene is void. Consequently $D_{PR}(V_1, V_3) = 1$. On the other hand $|V_1 \cap V_2| = |V_2 \cap V_3| = 1$ while $|V_1 \cup V_2| = |V_2 \cup V_3| = 2$. As a result $D_{PR}(V_1, V_2) = D_{PR}(V_2, V_3) = 1/3$ and $D_{PR}(V_1, V_2) + D_{PR}(V_2, V_3) < D_{PR}(V_1, V_3)$. So, the implicit solution spaces employed when using F_{PR}, as well as F_{CO}, are non-metric spaces.

D. Further experimental comparison of performance measures

1) Computational complexity: It can be deduced from the F_{PR} definition that the performance evaluation of a video segmentation involving N shots requires $O(N^2)$ operations. Note that these operations can be no more complex than a summation and a logical AND. On the other hand, in order to compute either F_{CO} or DED, the construction of the co-occurrence matrix is required. This matrix is built by sequentially browsing all shots of the video and thus requiring $O(N)$ operations. The co-occurrence matrix has a size of $|A_G| \times |A_E|$, where $|A_G|$ and $|A_E|$ is the number of scenes in the ground truth and the experimental segmentation, respectively. After its estimation, F_{CO} computation involves all co-occurrence matrix elements, but only linear combinations of them. So, the overall computational complexity of F_{CO} is $O(N) + O(|A_G| \cdot |A_E|)$.

Finally, DED also employs the co-occurrence matrix, which is decomposed into sub-videos using splitting boundaries. Consequently, the overall complexity is of $O(N) + O(DED)$ where $O(DED)$ is the complexity related to the total sub-video DED estimation. The theoretical determination of this computational complexity is not a trivial task, since it depends on the number of splitting boundaries, as well as the number of ground truth and experimentally estimated scenes. More specifically, if the $|A_G|$ ground truth boundaries are experimentally estimated with a Recall rate R and a Precision rate P, then the video will be divided into $R \cdot |A_G| + 1$ sub-videos. These sub-videos will include, in total, $(1 - R) \cdot |A_G|$ ground truth scene boundaries and $(1 - P) \cdot |A_E|$ experimental scene boundaries that are not sub-video boundaries as well. Typical values of Recall and Precision, as those given in [24], are significantly over 50%. If this baseline performance is assumed and $|A_E|$ and $|A_G|$ are assumed both equal to 40, then each sub-video would contain on average less than 1 ground truth and less than 1 experimentally estimated scene boundaries. So, in practical situations the DED algorithm computational complexity is expected not to be significantly higher than $O(N)$. But, it should be mentioned that the worse case complexity is higher than the one related to F_{CO}, since the job assignment complexity is cubic.

An experimental evaluation of the computational complexity of DED, F_{PR} and F_{CO} was carried out on the datasets of section IV-B, and the results (expressed as the ratio of F_{PR} or F_{CO} computation time over DEDs computation time) are given in Tables III, IV and V. These tables demonstrate the higher efficiency of the DED measure. The observed differences between the three datasets are explained by the fact that in the news and the movie datasets, the video streams comprise more shots, but are decomposed into fewer and longer ground truth scenes. Consequently, the F_{PR} computational cost, which is fully determined by the number of shots, increases, while the computational cost associated with the browsing of the co-occurrence matrix remains unaffected.

Method	[23]	[11]	[25]	[26]
F_{PR} / DED	1.1959	1.1229	1.2156	1.0506
F_{CO} / DED	9.6109	9.1970	6.9088	7.577

TABLE III
COMPUTATIONAL COST OF F_{PR} AND F_{CO} OVER DED IN THE DOCUMENTARY DATASET.

Method	[23]	[11]	[25]	[26]
F_{PR} / DED	5.133	3.1586	2.6934	2.8256
F_{CO} / DED	4.0909	2.5779	3.2347	3.3698

TABLE IV
COMPUTATIONAL COST OF F_{PR} AND F_{CO} OVER DED IN THE MOVIE DATASET.

Method	[23]	[11]	[25]	[26]
F_{PR} / DED	8.6751	8.34	8.6081	8.9471
F_{CO} / DED	2.9818	2.8892	2.557	2.6029

TABLE V
COMPUTATIONAL COST OF F_{PR} AND F_{CO} OVER DED IN THE NEWS DATASET.

The efficiency of DED is to a great extent due to the decomposition of the video to sub-videos (according to the method of section III-C). This can be demonstrated if DED's computation time is contrasted with the computation time of a DED variant that does not decompose the video to sub-videos. The corresponding results are shown in Table VI. As will be discussed in the next subsection, the computational complexity that is associated with the evaluation of the measure plays a critical role in the overall computation time that the parameter tuning of a scene segmentation technique would require.

2) Parameter sampling density: The parameters of a scene segmentation system, when no specific guidelines are available, are typically determined by search in the parameter space; this involves a uniform sampling of the parameter space [11]. This parameter tuning is conducted by varying a parameter value that generates an error signal, where the domain of the error signal is the parameter value space and

Dataset	Documentary	Movie	News
Non-optimized computation time / Optimized computation time	2.4964	11.9849	34.439

TABLE VI
COMPUTATION TIME WITHOUT DECOMPOSING THE VIDEO TO SUB-VIDEOS DIVIDED BY COMPUTATION TIME WHEN DECOMPOSING THE VIDEO TO SUB-VIDEOS ACCORDING TO SECTION III-C.

the values of the error signal are the distances of the resulting segmentations from the ground truth one. The latter distance is calculated using a segmentation evaluation measure. The computation time required for this process is affected not only by the computational complexity of the evaluation measures but also by the required parameter sampling density.

The minimum sampling density is determined by the Nyquist-Shannon sampling theorem as being proportional to the spectrum bandwidth of the error signal (i.e., assuming that it is a bandlimited signal, to its highest frequency). It should be noted that when a signal is multi-dimensional, i.e. more than one parameters are tuned at the same time, then the Nyquist-Shannon sampling theorem is applied separately in each different dimension. In order to determine the highest frequency, a thresholding is required, since in theory the spectrum of any signal limited in time is not limited in frequency. Instead of employing a strict, arbitrarily chosen threshold, we selected 20 different thresholds, varying from 0.1% of the total spectrum power to 2%, and averaged the results.

Furthermore, when conducting the experimental analysis, it is not the analog error signal that is taken into account but inevitably a digital approximation of it, which is generated using a manually chosen sampling rate. In order to prevent error signal aliasing, the sampling rate used to generate it should exceed the Nyquist-Shannon rate. This can not be theoretically guaranteed, since it would require a priori knowledge of the signal spectrum under examination. However, this problem may be circumvented by relying on the fact that when sampling exceeds the Nyquist-Shannon rate then the bandlimited spectrum is identical and independent from the sampling frequency. So, the adopted strategy was to double the sampling points until the spectra of all three approximate error signals e_{PR}, e_{CO}, e_{DED} stabilized. This strategy is summarized in Algorithm 2. It should be noted that the number of samples doubles (Step 5) before the termination control (Step 6) in order to provide extra accuracy to the spectra estimation.

The experimental setup was identical to the one employed for computational complexity, i.e. it included the four scene segmentation techniques and the three different datasets. The results (comparing the highest frequency of the error signal spectrum when using DED, F_{PR} and F_{CO}) are shown in Tables VII, VIII and IX. These tables show that the F_{PR}/DED or F_{CO}/DED bandwidth ratio is not so much dependent on the dataset, but rather on the employed scene segmentation technique. However, it can be seen that in all experiments, only on two occasions the sampling rate of the DED error signal was required to be greater than that of F_{CO}, while DED outperforms F_{PR} for all examined methods and datasets. Consequently, it can be concluded that by employing DED, the sampling required to tune the system parameters is more sparse than if F_{PR} or F_{CO} were employed. The total computational gain is estimated by multiplying the corresponding gain values from Tables III to IX. It can be seen that through the use of DED the scene segmentation tuning becomes much faster, with a speed up factor that reaches up to $10 - 15$ times.

Algorithm 2 Sampling Rate Estimation Summary

```
1: The error signals $e_{PR}$, $e_{CO}$, $e_{DED}$ are estimated for the $3$ different distance measures and for parameter values from $0$ to a maximum value $T$. The sampling rate is fixed to $T/R_0$. Quantity $R_0$, which determines the initial sampling rate, is a constant.
2: Initialization: $S = 1$, $f_{PR} = FFT(e_{PR})$, $f_{CO} = FFT(e_{CO})$, $f_{DED} = FFT(e_{DED})$.
3: $\lambda = T/(2^S \cdot R_0)$
4: The error signals are recomputed by estimating their values for the additional parameter values $(T \cdot i)/(2^{S-1} \cdot R_0) + \lambda$, $i = 0, 1, 2, ..., 2^{S-1} \cdot R_0 - 1$.
5: $S = S + 1$.
6: The FFTs of the error signals are re-estimated and compared to the corresponding $f$ variables. If all of them are similar to the corresponding $f$s, the algorithm terminates and the sampling is performed with rate $T/(2^S \cdot R_0)$. Else, the estimated FFTs become the new $f$s and the algorithm continues from Step 3.
```

Method	[23]	[11]	[25]	[26]
F_{PR} / DED	1.3511	1.1244	1.3173	1.5475
F_{CO} / DED	1.023	1.6635	1.7217	2.0594

TABLE VII
BANDWIDTH OF F_{PR} AND F_{CO} OVER DED IN THE DOCUMENTARY DATASET.

Method	[23]	[11]	[25]	[26]
F_{PR} / DED	1.2605	1.0671	1.089	1.431
F_{CO} / DED	0.923	1.7809	1.7685	1.7534

TABLE VIII
BANDWIDTH OF F_{PR} AND F_{CO} OVER DED IN THE MOVIE DATASET.

Method	[23]	[11]	[25]	[26]
F_{PR}/DED	1.0582	1.0653	1.0608	1.109
F_{CO}/DED	0.8438	1.1794	1.2316	1.3993

TABLE IX
BANDWIDTH OF F_{PR} AND F_{CO} OVER DED IN THE NEWS DATASET.

E. User Study

In addition to the above experiments, we conducted a user study involving 6 non-expert users in order to further assess how well the results of the proposed DED measure match the expectations of human evaluators. For the needs of this study we randomly produced triplets of synthetic segmentations for a subset of the videos of our datasets, and then selected 20 of those triplets for which the three considered evaluation measures disagree in the ranking of each triplet's segmentations (e.g. segmentation triplets for which DED suggests that the first segmentation is the most similar to the manually-created ground-truth one, while F_{PR} and F_{CO} suggest that the second and the third one are most similar to the ground truth, respectively). The 20 triples were shown one by one to a set of 6 non-expert users, who independently viewed the (segmented) videos and ranked each of them, without having any knowledge of the corresponding DED, F_{PR} and F_{CO} values. The agreement of the user rankings with the rankings generated by each measure was evaluated using normalized inversion count [27] and the results are shown in Table X. It can be seen that DED has significantly better (i.e., lower) scores than F_{PR} and F_{CO}.

Segmentation Evaluation Measure	DED	F_{PR}	F_{CO}
Normalized Inversion Count	0.16	0.37	0.53

TABLE X
RESULTS OF THE CONDUCTED USER STUDY. NORMALIZED INVERSION COUNT EXPRESSES HOW WELL THE OUTPUT OF EACH EVALUATION MEASURE AGREES WITH THE RESULTS OF HUMAN EVALUATORS (LOWER SCORES INDICATE BETTER AGREEMENT).

Finally, a few qualitative examples of scene segmentation evaluation are given in Fig. 4, illustrating the values of the F_{PR}, F_{CO} and DED measures in realistic scene segmentation cases. These examples further emphasize the superiority of the DED metric in producing evaluation results which are in better agreement with the human perception of segmentation goodness, compared to F_{PR} and F_{CO}.

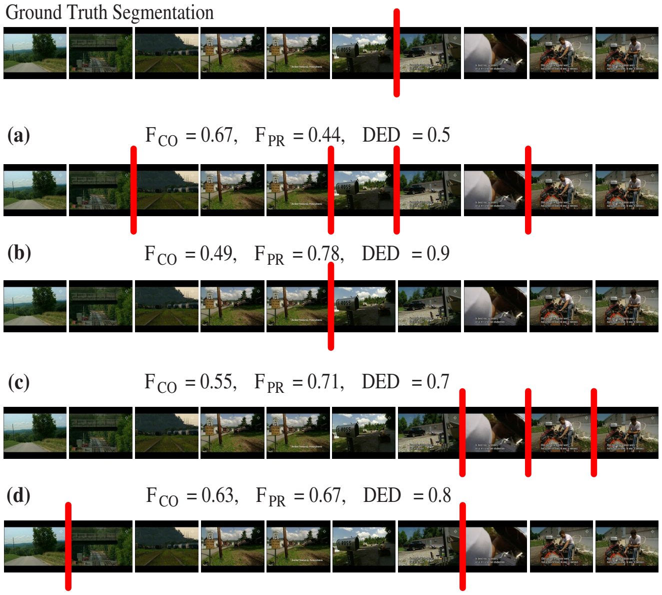
Ground Truth Segmentation
(a)
F$_{CO}$ = 0.67, F$_{PR}$ = 0.44, DED = 0.5
(b)
F$_{CO}$ = 0.49, F$_{PR}$ = 0.78, DED = 0.9
(c)
F$_{CO}$ = 0.55, F$_{PR}$ = 0.71, DED = 0.7
(d)
F$_{CO}$ = 0.63, F$_{PR}$ = 0.67, DED = 0.8

Fig. 4. An example of scene segmentation evaluation using F_{PR}, F_{CO} and DED. In each of the five rows 10 key-frames, belonging to 10 adjacent video shots, are presented. The vertical lines represent the scene boundaries (either ground truth boundaries or automatically detected ones). In the first row the ground-truth segmentation of the video is shown. The video includes two scenes, comprising 6 and 4 shots, respectively. In example result (a) the correct scene boundary and 3 additional false scene boundaries have been detected. Example result (b) only misplaces the scene boundary by 1 shot. Example result (c) misplaces the correct scene boundary by 1 shot and furthermore reports two false boundaries at the end of the video. Example result (d) also misplaces the correct scene boundary by 1 shot, and reports only one false boundary at the beginning of the video. It is expected that all evaluation measures would consider example result (b) as being better than (a), and (d) being better than (c). However, the F_{CO} of (a) is higher than that of (b) and the F_{PR} of (c) is higher than that of (d). On the other hand, DED manages to evaluate these results according to what is intuitively expected.

V. CONCLUSION

In this work a novel scene segmentation evaluation measure was presented. Furthermore, an implementation that computes this measure with less than cubic complexity was introduced. For testing the metric's ability to model efficiently the human performance rating, a number of required measure properties were introduced. The proposed measure and two baseline performance measures were comparatively evaluated with respect to their compliance with these properties. Furthermore, an experimental setup was used to examine the computational cost that is associated with the parameter tuning of a scene segmentation system, when this process is guided by one of these evaluation measures. These results, together with the results of a small user study that was also conducted, demonstrate that the presented measure outperforms those currently employed in the literature and provides an efficient approach to comparing automatic scene segmentation techniques and to guiding the optimization of their parameters. The software implementation of DED is available at http://mklab.iti.gr/project/ded.

REFERENCES

[1] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello, "Foveated shot detection for video segmentation," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 365–377, March 2005.

[2] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–588, June 1999.

[3] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 411–418, April 2010.

[4] C. J. van Rijsbergen, *Information Retrieval*. Butterworth-Heinemann, London, 2nd edition, 1979.

[5] J. Vendrig and M. Worring, "Systematic evaluation of logical story unit segmentation," *IEEE Trans. on Multimedia*, vol. 4, no. 4, pp. 492–499, December 2002.

[6] W. Tavanapong and J. Zhou, "Shot clustering techniques for story browsing," *IEEE Trans. on Multimedia*, vol. 6, no. 4, p. 517527, August 2004.

[7] S. Benini, L.-Q. Xu, and R. Leonardi, "Identifying video content consistency by vector quantization," in *Proc. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Montreux, Switzerland, April 2005.

[8] Y. Zhao, T. Wang, P. Wang, W. Hu, Y. Du, Y. Zhang, and G. Xu, "Scene segmentation and categorization using ncuts," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, USA, June 2007, p. 17.

[9] V. Parshin, A. Paradzinets, and L. Chen, "Multimodal data fusion for video scene segmentation," in *Proc. Int. Conf. on Visual Information and Information Systems*, Amsterdam, The Netherlands, July 2005, p. 279289.

[10] K. Wilson and A. Divakaran, "Discriminative genre-independent audio-visual scene change detection," in *Proc. SPIE Conf. on Multimedia Content Access: Algorithms and Systems III*, vol. 7255, San Jose, CA, USA, January 2009.

[11] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho,

and I. Trancoso, “Temporal video segmentation to scenes using high-level audiovisual features,” *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1163–1177, August 2011.

[12] V. Chasanis, A. Likas, and N. Galatsanos, “Scene Detection in Videos Using Shot Clustering and Sequence Alignment,” *IEEE Trans. on Multimedia*, vol. 11, no. 1, pp. 89–100, January 2009.

[13] D. Mitrovic, S. Hartlieb, M. Zeppelzauer, and M. Zaharieva, “Scene Segmentation in Artistic Archive Documentaries,” in *Proc. 6th Symp. of the Workgroup Human-Computer Interaction and Usability Engineering (USAB)*, Klagenfurt, 2010, pp. 400–410.

[14] H. Chen and C. Li, “A Practical Method for Video Scene Segmentation,” in *Proc. 3rd IEEE Int. Conf. on Computer Science and Information Technology (ICCSIT)*, Chengdu, China, July 2010, pp. 153–156.

[15] S. Zhu and Y. Liu, “Video scene segmentation and semantic representation using a novel scheme,” *Multimedia Tools and Applications*, vol. 42, no. 2, pp. 183–205, April 2009.

[16] L. Duan, D. Xu, I. W. Tsang, and J. Luo, “Visual Event Recognition in Videos by Learning from Web Data,” in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010, pp. 1959–1966.

[17] D. Xu, T. J. Cham, S. Yan, L. Duan, and S.-F. Chang, “Near Duplicate Identification with Spatially Aligned Pyramid Matching,” *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 20, no. 8, pp. 1068–1079, August 2010.

[18] R. W. Hamming, “Error detecting and error correcting codes,” *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[19] S. Koknar-Tezel and L. J. Latecki, “Improving SVM Classification on Imbalanced Data Sets in Distance Spaces,” in *Proc. Ninth IEEE Int. Conf. on Data Mining (ICDM)*, Miami, FL, USA, December 2009, pp. 259–267.

[20] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, 2000.

[21] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, pp. 2319–2323, 2000.

[22] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[23] M. Yeung, B.-L. Yeo, and B. Liu, “Segmentation of video by clustering and graph analysis,” *Computer Vision and Image Understanding*, vol. 71, no. 1, pp. 94–109, July 1998.

[24] Z. Rasheed and M. Shah, “Detection and Representation of Scenes in Videos,” *IEEE Trans. on Multimedia*, vol. 7, no. 6, pp. 1097–1105, December 2005.

[25] N. Nitanda, M. Haseyama, and H. Kitajima, “Audio signal segmentation and classification for scene-cut detection,” in *IEEE Int. Symp. on Circuits and Systems*, vol. 4, April 2005, pp. 4030–4033.

[26] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, and I. Trancoso, “Multi-modal scene segmentation using scene transition graphs,” in *Proc. ACM Multimedia*, Beijing, China, October 2009, pp. 665–668.

[27] A. Tiskin, “Semi-local String Comparison: Algorithmic Techniques and Applications,” *Mathematics in Computer Science*, vol. 1, no. 4, pp. 571–603, 2008.