Measuring Scene Detection Performance

Lorenzo Baraldi, Costantino Grana, Rita Cucchiara

Dipartimento di Ingegneria "Enzo Ferrari"
Università degli Studi di Modena e Reggio Emilia
Via Vivarelli 10, Modena MO 41125, Italy
name.surname@unimore.it

Abstract. In this paper we evaluate the performance of scene detection techniques, starting from the classic precision/recall approach, moving to the better designed coverage/overflow measures, and finally proposing an improved metric, in order to solve frequently observed cases in which the numeric interpretation is different from the expected results. Numerical evaluation is performed on two recent proposals for automatic scene detection, and comparing them with a simple but effective novel approach. Experimental results are conducted to show how different measures may lead to different interpretations.

Keywords: scene detection, measures, clustering.

1 Introduction

The large availability of videos on the Internet has led to great interest in fields different from simple entertainment or news broadcasting, such as education (Massive Open Online Courses). This also led to a strong interest in the re-use of video content coming from major broadcasting networks, which have been producing high quality edited videos for popular science purposes.

Unfortunately, re-using videos in ones own presentations or video aided lectures is not an easy task, and requires video editing skills and tools. There is a growing need for managing video content, but the basic unit for this task cannot be the single frame: higher level groupings are needed, such as DVD chapters. The problem is that most of the on-line reusable content is not provided with editor defined video sub units. Scene detection may help in this situation, going beyond frames and even beyond simple editing units, such as shots [3]. The task is to identify coherent sequences (scenes) in videos, without any help from the editor or publisher. As it is common in newer research areas, evaluating the performance of automatic systems is not an easy task [2]: techniques previously employed for different purposes are applied to newer problems, even if they do not perfectly match with the objective at hand, but are easily understood from previous experience. Often this approach leads to erroneous interpretations of the experimental evaluations.

In this paper we try to tackle the problem of evaluating scene detection techniques, starting from the classic precision/recall approach, moving to the better designed coverage/overflow measures, and finally propose an improved

definition of the latter ones, which solve frequently observed cases in which the numeric interpretation would be quite different from the expected results. Numerical evaluation is performed on two recent proposals for automatic scene detection, which are compared with the different measures, together with our simple approach. The experimental results will show the different aspects which may be wrongly evaluated with unsuitable measures.

2 Recent Scene Detection techniques

Video decomposition techniques aim to partition a video into sequences, like shots or scenes. Shots are elementary structural segments that are defined as sequences of images taken without interruption by a single camera. Scenes, on the contrary, are often defined as series of temporally contiguous shots characterized by overlapping links that connect shots with similar content [5]. Most of the existing works can be roughly categorized into three categories: *rule-based methods*, that consider the way a scene is structured in professional movie production, *graph-based methods*, where shots are arranged in a graph representation, and *clustering-based methods*. They can rely on visual, audio, and textual features.

We focus our evaluation on three different scene detection algorithms. We propose a clustering approach, where we modify the standard spectral clustering algorithm in order to produce temporally consistent clusters; we evaluate the method in [4], where scene boundaries are detected from the alignment score of symbolic sequences, and the multimodal approach presented in [7].

A spectral clustering approach Our scene detection method generates scenes by grouping adjacent shots. Shots are described by means of color histograms, hence relying on visual features only: given a video, we compute a three-dimensional histogram for each frame, by quantizing each RGB channel in eight bins, for a total of 512 bins. Then, we sum histograms from frames belonging to the same shot, thus obtaining a single L_1-normalized histogram for each shot.

In contrast to other approaches that used spectral clustering for scene detection, we build a similarity matrix that jointly describes appearance similarity and temporal proximity. Its generic element κ_{ij} defines the similarity between shots $\mathbf{x}_i$ and $\mathbf{x}_j$ as

$$\kappa_{ij} = \exp\left(-\frac{d_1^2(\psi(\mathbf{x}_i), \psi(\mathbf{x}_j)) + \alpha \cdot d_2^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right) \tag{1}$$

where $\psi(\mathbf{x}_i)$ is the normalized histogram of shot $\mathbf{x}_i$, d_1^2 is the Bhattacharyya distance and $d_2^2(\mathbf{x}_i, \mathbf{x}_j)$ is the normalized temporal distance between shot $\mathbf{x}_i$ and shot $\mathbf{x}_j$, while the parameter α tunes the relative importance of color similarity and temporal distance. To describe temporal distance between frames, $d_2^2(\mathbf{x}_i, \mathbf{x}_j)$ is defined as

$$d_2^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{|m_i - m_j|}{l} \tag{2}$$

where m_i is the index of the central frame of shot $\mathbf{x}_i$, and l is the total number of frames in the video. The spectral clustering algorithm is then applied to the

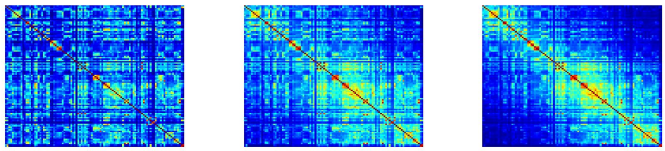


Fig. 1. Effect of α (from left to right 0, 0.5 and 1) on similarity matrix κ_{ij}. Higher values of α enforce connections between near shots and increase the quality of the detected scenes (best viewed in color).

similarity matrix, using the Normalized Laplacian and the maximum eigen-gap criterion to select k, that therefore is equal to $\arg\max |\lambda_i - \lambda_{i-1}|$, where λ_i is the i-th eigenvalue of the Normalized Laplacian.

As shown in Fig. 1, the effect of applying increasing values of α to the similarity matrix is to raise the similarities of adjacent shots, therefore boosting the temporal consistency of the resulting groups. Of course, this does not guarantee a completely temporal consistent clustering (i.e. some clusters may still contain non-adjacent shots); at the same time, too high values of α would lead to a segmentation that ignores color dissimilarity. The final scene boundaries are created between adjacent shots that do not belong to the same cluster.

A sequence alignment approach The method presented in [4], unlike the previous one, represents shots by means of key-frames. The first step of this method, therefore, is to extract several key-frames from each shot: frames from a shot are clustered using the spectral clustering algorithm, color histograms as features, and the euclidean distance to compute the similarity matrix. The number of clusters is selected by applying a threshold Th on the eigenvalues of the Normalized Laplacian.

The distance between a pair of shots is defined as the maximum similarity between key-frames belonging to the two shots, computed using histogram intersection. Shots are clustered using again spectral clustering and the aforesaid distance measure, and then labeled according to the clusters they belong to. The same threshold Th is used to select the number of clusters at this step.

Finally, to create scene boundaries, they compare successive non-overlapping windows of shot labels using a modified version of the Needleman-Wunsh algorithm, that considers the visual similarity of shot clusters and the frequency of sequential labels in the video.

A multimodal technique The method in [7] extends the Shot Transition Graph (STG) using multimodal low-level and high-level features. To this aim, multiple STGs are constructed, one for each kind of feature, and then a probabilistic merging process is used to combine their results.

The used features include visual features, such as HSV histograms, outputs of visual concept detectors trained using the Bag of Words approach, and audio features, like background conditions, speaker histogram, and model vectors constructed from the responses of a number of audio event detectors.

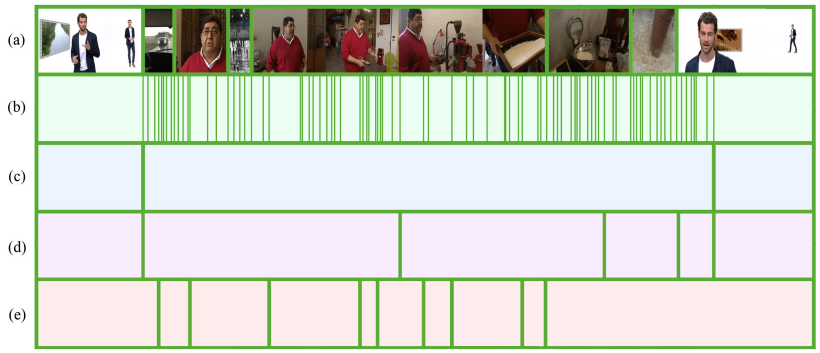


Fig. 2. Samples results on our dataset. Row (a) shows the ground-truth segmentation, (b) the individual shots boundaries, row (c) shows the results of our method, (d) those of [7] and (e) those of [4] (best viewed in color).

3 Measures for evaluating scene segmentation

To evaluate the results of the aforementioned approaches, we organize evaluation measures in three categories: *boundary-level measures*, that consider the problem of scene detection as a boundary detection problem, and therefore evaluate correctly and wrongly detected boundaries; *shot-level measures* that, on the contrary, compare the number of corresponding or overlapping shots between the ground truth and the detected segmentation, and *frame-level measures*, that consider the number of frames instead of the number of shots.

Boundary level The first level to assess the quality of a scene segmentation is to count correctly and wrongly detected boundaries, without considering the temporal distance between a ground truth cut and the nearest detected cut. The most used measures in this context are precision and recall, together with the F-Score measure, that summarizes both. Precision is the ratio of the number of correctly identified scenes boundaries to the total number of scenes detected by the algorithm. Recall is the ratio of the number of correctly identified boundaries to the total number of scenes in the ground truth.

Of course this kind of evaluation does not discern the seriousness of an error: if a boundary is detected one shot before or after its ground truth position, an error is counted in recall as if the boundary was not detected at all, and in precision as if the boundary was put far away. This issue appears to be felt also by other authors, with the result that sometimes a tolerance factor is used. For example, [6] uses a *best match* method with a sliding window of 30 seconds, so that a detected boundary is considered correct if it matches a ground truth boundary in the sliding window.

Shot level On an different level, detected scene can be evaluated with regards to their compliance to the ground truth in terms of overlap. Vendrig *et al.* [8], for example, proposed the Coverage and Overflow measures. Coverage $\mathcal{C}$ measures the quantity of shots belonging to the same scene correctly grouped together, while Overflow $\mathcal{O}$ evaluates to what extent shots not belonging to the same scene are erroneously grouped together. Formally, given the set of automat-

ically detected scenes $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_m]$, and the ground truth $\tilde{\mathbf{s}} = [\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, ..., \tilde{\mathbf{s}}_n]$, where each element of $\mathbf{s}$ and $\tilde{\mathbf{s}}$ is a set of shot indexes, the coverage $\mathcal{C}_t$ of scene $\tilde{\mathbf{s}}_t$ is proportional to the longest overlap between $\mathbf{s}_i$ and $\tilde{\mathbf{s}}_t$:

$$\mathcal{C}_t = \frac{\max_{i=1...,m} \#(\mathbf{s}_i \cap \tilde{\mathbf{s}}_t)}{\#(\tilde{\mathbf{s}}_t)} \tag{3}$$

where $\#(\mathbf{s}_i)$ is the number of shots in scene $\mathbf{s}_i$. The overflow of a scene $\tilde{\mathbf{s}}_t$, $\mathcal{O}_t$, is the amount of overlap of every $\mathbf{s}_i$ corresponding to $\tilde{\mathbf{s}}_t$ with the two surrounding scenes $\tilde{\mathbf{s}}_{t-1}$ and $\tilde{\mathbf{s}}_{t+1}$:

$$\mathcal{O}_t = \frac{\sum_{i=1}^{m} \#(\mathbf{s}_i \setminus \tilde{\mathbf{s}}_t) \cdot \min(1, \#(\mathbf{s}_i \cap \tilde{\mathbf{s}}_t))}{\#(\tilde{\mathbf{s}}_{t-1}) + \#(\tilde{\mathbf{s}}_{t+1})} \tag{4}$$

The computed per-scene measures can then be aggregated into values for an entire video as follows:

$$\mathcal{C} = \sum_{t=1}^{n} \mathcal{C}_t \cdot \frac{\#(\tilde{\mathbf{s}}_t)}{\sum \#(\tilde{\mathbf{s}}_i)}, \quad \mathcal{O} = \sum_{t=1}^{n} \mathcal{O}_t \cdot \frac{\#(\tilde{\mathbf{s}}_t)}{\sum \#(\tilde{\mathbf{s}}_i)} \tag{5}$$

finally, an F-Score measure can be defined to combine Coverage and Overflow, by taking the harmonic mean of $\mathcal{C}$ and $1 - \mathcal{O}$.

Frame level We identify two drawbacks of Vendrig's measures, hence propose an improved definition of these. The first one is that, being computed at the shot level, an error on a short shot is given the same importance of an error on a very long shot. On the other hand, we propose to normalize $\mathcal{O}_t$ with respect to the length of $\tilde{\mathbf{s}}_t$ instead of that of $\tilde{\mathbf{s}}_{t-1}$ and $\tilde{\mathbf{s}}_{t+1}$, since we believe that the amount of error due to overflowing should be related to the current scene length, instead of its two neighbors. As an example, consider a ground truth segmentation where a long scene is surrounded by two short scenes: if the detected scene is the union of all three, the actual amount of overflow for the middle scene is quite small, while the usage of shot-level measures would result in a 100% overflow.

Therefore, we propose the Coverage* and Overflow* metrics, where the cardinality operator # is replaced with the number of frames of a scene, $l(\mathbf{s}_i)$, and overflow is redefined as follows:

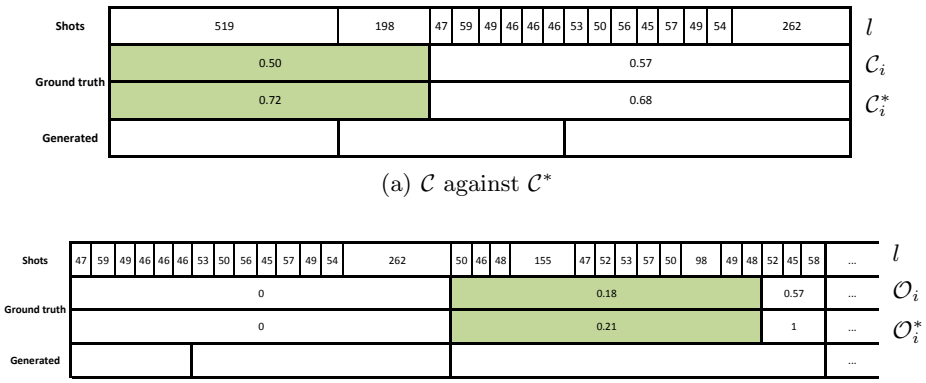
$$\mathcal{O}_t^* = \min\left(1, \frac{\sum_{i=1}^{m} l(\mathbf{s}_i \setminus \tilde{\mathbf{s}}_t) \cdot \min(1, l(\mathbf{s}_i \cap \tilde{\mathbf{s}}_t))}{l(\tilde{\mathbf{s}}_t)}\right) \tag{6}$$

Note that we limit the amount of overflow to one. The corresponding $\mathcal{C}^*$ and $\mathcal{O}^*$ for an entire video can be obtained in the same way of Eq. 5, using the newly defined cardinality operator.

4 Evaluation

We evaluate the aforesaid measures and algorithms on a collection of ten challenging broadcasting videos from the Rai Scuola video archive[1], mainly documentaries and talk shows. Shots have been obtained running the state of the

[1] `http://www.scuola.rai.it`



(a) $\mathcal{C}$ against $\mathcal{C}^*$

(b) $\mathcal{O}$ against $\mathcal{O}^*$

Fig. 3. Comparison of shot level and frame level measures.

art shot detector of [1] and manually grouped into scenes by a set of human experts to define the ground truth. Our dataset and the corresponding annotations are available for download at `http://imagelab.ing.unimore.it/files/RaiSceneDetection.zip`. We reimplemented the approach in [4] and used the executable of [7] provided by the authors. The threshold Th of [4] was selected to maximize the performance on our dataset, and α was set to 0.05 in all our experiments.

Tables 1, 2, and 3 compare the three different approaches using Boundary level, Shot level and Frame level metrics. As show in Table 1, detected boundaries rarely correspond to ground truth boundaries exactly, therefore leading to poor results in terms of precision and recall, even when considering a recent and state-of-the-art approach like [7]. The difference between the results obtained with shot and frame level measures, on the other hand, are produced by the alteration of the cardinality operator and by the change of normalization in Overfow*.

To visualize the effect of our improved definition of coverage, consider Fig. 3(a), where we compare the two definitions of coverage on a frame sequences from our dataset. First row shows the detected shots and their corresponding length in frames, while the second and the third rows show the ground truth and generated scene segmentation. The first ground truth scene, according to Vendrig's definition, gets a 0.5 coverage, since the generated scene covers one shot out of two. Our definition, on the contrary, considers the number of frames inside a shot, and therefore accounts for the fact the first shot is longer than the second. This results in a 0.72 coverage, which is surely a more realistic numerical result.

In Figure 3(b), instead, we compare Overflow and Overflow*. As it can be seen, the overflow of the first ground truth scene is zero according to both measures, since the corresponding generated scenes don't overlap with others ground truth scenes, while the numerical results for the second ground truth scenes are quite similar, even if Overflow* considers the number of frames and has a different kind of normalization. The difference between our definition and Vendrig's one becomes clear in the third ground truth scene, where our measure reports

Table 1. Performance comparison using Boundary level metrics.

Video	Spectral Clustering			Chasanis *et al.* [4]			Sidiropoulos *et al.* [7]		
	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall
V_1	0.12	0.09	0.17	0.25	0.20	0.33	**0.29**	0.25	0.33
V_2	**0.36**	0.27	0.55	0.00	0.00	0.00	0.30	0.33	0.27
V_3	**0.37**	0.29	0.53	0.13	0.13	0.13	0.31	0.36	0.27
V_4	**0.30**	0.23	0.43	0.10	0.10	0.10	0.22	0.50	0.14
V_5	**0.44**	0.31	0.75	0.00	0.00	0.00	0.36	0.31	0.42
V_6	0.18	0.10	0.75	0.00	0.00	0.00	**0.36**	0.29	0.50
V_7	**0.18**	0.33	0.13	0.00	0.00	0.00	0.13	0.13	0.13
V_8	0.10	0.06	0.27	0.13	0.10	0.18	**0.21**	0.25	0.18
V_9	**0.25**	0.16	0.62	0.00	0.00	0.00	0.21	0.33	0.15
V_{10}	0.23	0.15	0.60	**0.26**	0.38	0.20	0.19	0.33	0.13
Average	0.25	0.20	0.48	0.09	0.09	0.09	**0.26**	0.31	0.25

Table 2. Performance comparison using Shot level metrics.

Video	Spectral Clustering			Chasanis *et al.* [4]			Sidiropoulos *et al.* [7]		
	F-Score	$\mathcal{C}$	$\mathcal{O}$	F-Score	$\mathcal{C}$	$\mathcal{O}$	F-Score	$\mathcal{C}$	$\mathcal{O}$
V_1	0.64	0.81	0.48	0.70	0.64	0.24	**0.72**	0.84	0.37
V_2	**0.68**	0.61	0.22	0.36	0.80	0.77	0.59	0.85	0.55
V_3	**0.65**	0.68	0.38	0.58	0.73	0.52	0.58	0.90	0.57
V_4	**0.74**	0.69	0.22	0.50	0.65	0.60	0.33	0.94	0.80
V_5	**0.77**	0.68	0.11	0.25	0.93	0.86	0.66	0.76	0.41
V_6	0.51	0.37	0.17	0.18	0.89	0.90	**0.71**	0.77	0.34
V_7	0.30	0.97	0.82	0.37	0.70	0.75	**0.51**	0.78	0.62
V_8	0.59	0.53	0.33	**0.62**	0.57	0.32	0.45	0.88	0.70
V_9	**0.67**	0.55	0.15	0.27	0.87	0.84	0.43	0.92	0.72
V_{10}	**0.57**	0.42	0.12	0.54	0.91	0.62	0.44	0.94	0.71
Average	**0.61**	0.63	0.30	0.44	0.77	0.64	0.54	0.86	0.58

a 100% overflow, since the third generated scene overlaps the ground truth one by more than its length, while Vendrig's definition only reports a 0.57 overflow, since the next scene, which has no intersection with the detected one, is 9 shots long.

Finally, we note that the three metrics behave differently and there is not a complete agreement among them: [4] performs worse than the other two methods according to all the three measures, while [7] performs equal or slightly worse than our spectral clustering approach according to Boundary level and Shot level metrics. When shot duration is taken into account, using Frame level metrics, our spectral clustering approach considerably outperforms all the others approaches.

5 Conclusions

We have investigated the problem of evaluating scene detection algorithms and suggested metrics that try to reduce the gap between the numerical evaluation and the expected qualitative results. Experiments have been conducted on three different groups of metrics and on three different and recent approaches to scene

Table 3. Performance comparison using the Frame level metrics.

Video	Spectral Clustering			Chasanis *et al.* [4]			Sidiropoulos *et al.* [7]		
	F-Score*	$\mathcal{C}^*$	$\mathcal{O}^*$	F-Score*	$\mathcal{C}^*$	$\mathcal{O}^*$	F-Score*	$\mathcal{C}^*$	$\mathcal{O}^*$
V_1	0.69	0.82	0.40	**0.70**	0.65	0.24	**0.70**	0.63	0.20
V_2	**0.76**	0.77	0.24	0.60	0.91	0.55	0.61	0.73	0.47
V_3	**0.69**	0.77	0.37	0.51	0.87	0.64	0.51	0.89	0.64
V_4	**0.68**	0.70	0.34	0.54	0.70	0.56	0.22	0.95	0.88
V_5	**0.77**	0.68	0.13	0.34	0.92	0.79	0.57	0.66	0.50
V_6	0.58	0.42	0.06	0.20	0.89	0.88	**0.74**	0.72	0.24
V_7	0.39	0.95	0.76	0.37	0.75	0.76	**0.56**	0.69	0.53
V_8	**0.63**	0.66	0.40	0.59	0.65	0.47	0.15	0.89	0.92
V_9	**0.77**	0.70	0.14	0.07	0.83	0.96	0.15	0.94	0.92
V_{10}	**0.65**	0.53	0.15	0.50	0.93	0.66	0.11	0.93	0.94
Average	**0.66**	0.70	0.30	0.44	0.81	0.65	0.43	0.80	0.63

segmentation. Results shows that the problem of scene detection is still far from being solved, and that simple approaches like our spectral clustering technique can sometimes achieve better or equivalent results than more complex methods.

Acknowledgments This work was carried out within the project "Città educante" (CTN01_00034_393801) of the National Technological Cluster on Smart Communities cofunded by the Italian Ministry of Education, University and Research - MIUR.

References

1. Apostolidis, E., Mezaris, V.: Fast Shot Segmentation Combining Global and Local Visual Descriptors. In: IEEE Int. Conf. Acoustics, Speech and Signal Process. pp. 6583–6587 (2014)
2. Baraldi, L., Paci, F., Serra, G., Benini, L., Cucchiara, R.: Gesture recognition in ego-centric videos using dense trajectories and hand segmentation. In: Proc. of 10th IEEE Embedded Vision Workshop (EVW). Columbus, Ohio (Jun 2014)
3. Bertini, M., Del Bimbo, A., Serra, G., Torniai, C., Cucchiara, R., Grana, C., Vezzani, R.: Dynamic Pictorially Enriched Ontologies for Video Digital Libraries. IEEE MultiMedia 16(2), 41–51 (Apr 2009)
4. Chasanis, V.T., Likas, C., Galatsanos, N.P.: Scene detection in videos using shot clustering and sequence alignment. IEEE Trans. Multimedia 11(1), 89–100 (2009)
5. Hanjalic, A., Lagendijk, R.L., Biemond, J.: Automated high-level movie segmentation for advanced video-retrieval systems. IEEE Trans. Circuits Syst. Video Technol. 9(4), 580–588 (1999)
6. Rasheed, Z., Shah, M.: Detection and representation of scenes in videos. IEEE Trans. Multimedia 7(6), 1097–1105 (2005)
7. Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., Trancoso, I.: Temporal video segmentation to scenes using high-level audiovisual features. IEEE Trans. Circuits Syst. Video Technol. 21(8), 1163–1177 (2011)
8. Vendrig, J., Worring, M.: Systematic evaluation of logical story unit segmentation. IEEE Trans. Multimedia 4(4), 492–499 (2002)