

An approach for video cut detection using bipartite graph matching as dissimilarity distance

Silvio Jamil F. Guimarães Zenilton K.G. do Patrocínio Jr.
Hugo Bastos de Paula

Pontifícia Universidade Católica de Minas Gerais (PUC Minas)
Rua Walter Ianni, 255 - São Gabriel - 31980-110 - Belo Horizonte - MG - Brazil
{sjamil,zenilton,hugo}@pucminas.br

Abstract

The video segmentation problem consists in the identification of the boundary between consecutive shots. When two consecutive frames are similar, they are considered to be in the same shot. In this work, we use the maximum cardinality of the bipartite graph matching between two frames as the dissimilarity distance in order to identify the cut locations. Thus, if two frames are similar then the maximum cardinality is high. We present some experiments to show the high performance of this distance.

1 Introduction

The video segmentation problem consists in the identification of the boundary between consecutive shots, called transition. The simplest transition between two consecutive shots is the sharp transition (cut) that is simply a concatenation of these shots. The common approach to cope with cut detection is based on the use of a dissimilarity measure. [10] and [5] review some of the most popular methods for cut detection, such as pixel-wise comparison, histogram comparison, etc. Unfortunately, cut detection is complicated by the presence of effects, like gradual transitions, flashes and fast camera and object motions [4, 12, 11].

Another approach to the video segmentation problem is to transform the video into a 2D image, and to apply image processing methods on this image to extract the different patterns related to each transition. This approach can be found in [9, 2, 6, 3, 1].

In this work, we propose a modified approach to cope with cut detection, in which, we replace the LCS computation [1] by the maximum cardinality of bipartite graph matching [7]. The main contribution of our work is the application of a simple distance, which is

invariant to rotation and translation, to solve a problem of video segmentation.

This paper is organized as follows. In Sec. 2 we define the bipartite graph matching. In Sec. 3 we present a methodology for cut detection. In Sec. 4 we perform a comparative analysis for cut detection involving our method and some other methods, using three different quality measures. According to these measures, we can verify that our method presents generally the best results. Some conclusions and a summary of future works are given in Sec. 5.

2 Bipartite graph matching

Let $\mathbb{A} \subset \mathbb{N}^2$, $\mathbb{A} = \{0, \dots, H-1\} \times \{0, \dots, W-1\}$, where H and W are the width and height of each frame, respectively, and, $\mathbb{T} \subset \mathbb{N}$, $\mathbb{T} = \{0, \dots, N-1\}$, in which N is the length of a video.

Definition 2.1 (Frame) A frame f is a function from \mathbb{A} to \mathbb{Z} , where for each spatial position (x, y) in \mathbb{A} , $f(x, y)$ represents the grayscale value at pixel location (x, y) .

Definition 2.2 (Video) A video V_N , in domain $2\mathbb{D} \times \mathbb{T}$, can be seen as a sequence of frames f . It can be described by

$$V_N = (f)_{t \in \mathbb{T}} \quad (1)$$

where N is the number of frames contained in the video.

Definition 2.3 (Visual rhythm ([2]) or spatio-temporal slice ([6])) Let $V_N = (f)_{t \in \mathbb{T}}$ be an arbitrary video, in domain $2\mathbb{D} \times \mathbb{T}$. The visual rhythm, in domain $1\mathbb{D} \times t$, is a simplification of the video in which each frame f_t is transformed into a vertical line of the visual rhythm image VR , defined by $VR(t, z) = f_t(r_x \times z + a, r_y \times z + b)$, where $z \in \{0, \dots, H_{VR} - 1\}$ and $t \in \{0, \dots, N-1\}$, H_{VR} and N are the height and

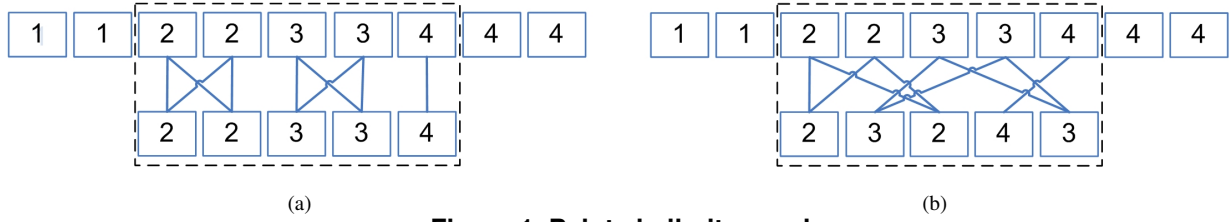


Figure 1. Point similarity graph

the width of the visual rhythm, respectively, r_x and r_y are ratios of pixel sampling, a and b are shifts on each frame.

The visual rhythm by sub-sampling (or simply visual rhythm) is a simplification of the video content represented by a 2D image. This simplification can be obtained by a systematic sampling of points of the video, such as, extraction of the diagonal points of each frame. Thus, according to these parameters, different pixel samplings could be considered, for example, if $r_x = r_y = 1$ and $a = b = 0$ and $H = W$ then we obtain all pixels of the principal diagonal. The choice of the pixel sampling constitutes a problem in the sense that different samplings produce different visual rhythms in which video events (cuts, fades, flashes, etc) will appear as different patterns. [2] presented different pixel sampling possibilities with their correspondent visual rhythms. They said that the best results are found when the sampling is based on a diagonal because it contains both horizontal and vertical features.

Based on those definitions, we define *point similarity* as follows.

Definition 2.4 (Point similarity) Let P_{l_1} and P_{l_2} be two points at location l_1 and l_2 , respectively. Two points are similar if a distance measure $\mathcal{D}(P_{l_1}, P_{l_2})$ between them is smaller than a specified threshold (δ). The point similarity is defined as

$$PS(P_{l_1}, P_{l_2}, \delta) = \begin{cases} 1, & \text{if } \mathcal{D}(P_{l_1}, P_{l_2}) \leq \delta \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The distance measure between two points ($\mathcal{D}(P_{l_1}, P_{l_2})$) is obtained from the Euclidean distance between their values. After that, it is possible to construct a point similarity graph based on two frames f_{t_1} and f_{t_2} . In our approach, these frames are represented by the columns t_1 and t_2 of the visual rhythm. Thus, we will compute the point similarity graph directly from the visual rhythm instead of the frames. It is important to note that for this approach we reduce the number of points and the computational time without decreasing the global performance.

Definition 2.5 (Point similarity graph – G^δ) Let f_{t_1} and f_{t_2} be two frames with M points. A point similarity graph $G^\delta = (N^{f_{t_1}} \cup N^{f_{t_2}}, E^\delta)$ is a bipartite graph. Each node $v^{f_{t_1}} \in N^{f_{t_1}}$ represents a point $P_{t_1} \in f_{t_1}$ and each node $v^{f_{t_2}} \in N^{f_{t_2}}$ represents a point $P_{t_2} \in f_{t_2}$. There is an edge $e \in E^\delta$ between $v^{f_{t_1}}$ and $v^{f_{t_2}}$ if frame similarity of associated frames is equal to 1, i.e.,

$$E^\delta = \{ (v^{f_{t_1}}, v^{f_{t_2}}) \mid v^{f_{t_1}} \in N^{f_{t_1}}, v^{f_{t_2}} \in N^{f_{t_2}}, PS(P_{t_1}, P_{t_2}, \delta) = 1 \} \quad (3)$$

As illustrated in Fig. 1, we match two frames with the same size (number of points). In this paper, we focus on cut transition detection. To do this, we define *matching* and *maximum cardinality matching* as follows.

Definition 2.6 (Matching – M_k^δ [7]) Let $G^\delta = (N^{f_{t_1}} \cup N^{f_{t_2}}, E^\delta)$ be a point similarity graph. A subset $M_k^\delta \subseteq E^\delta$ is a match if any two edges in M_k^δ are not adjacent.

Definition 2.7 (Maximum cardinality matching – \overline{M}_k^δ [7]) Let \overline{M}_k^δ be a matching in a point similarity graph G^δ . So, \overline{M}_k^δ is the maximum cardinality matching if there is no other matching M_k^δ in G^δ such that $|\overline{M}_k^\delta| > |M_k^\delta|$.

Finally, cut transition detection problem can be defined.

Definition 2.8 (Cut transition localization – CTL) The cut transition localization (CTL) problem corresponds to the identification of all sudden change between two consecutive frames. Thus, a frame f_{t_1} with M points that does not match to a frame f_{t_2} can be defined by

$$CTL(f_{t_1}, f_{t_2}, \delta, \Delta) = \{k \in \mathbb{T} \mid |\overline{M}_k^\delta| \leq \Delta\} \quad (4)$$

where \overline{M}_k^δ is the maximum cardinality matching of a point similarity graph G^δ which is generated using two frames f_{t_1} , f_{t_2} , and two specified thresholds δ and Δ . δ corresponds to Euclidean distance between two point values and Δ corresponds to the maximum matching that are permitted to classify the location as cut.

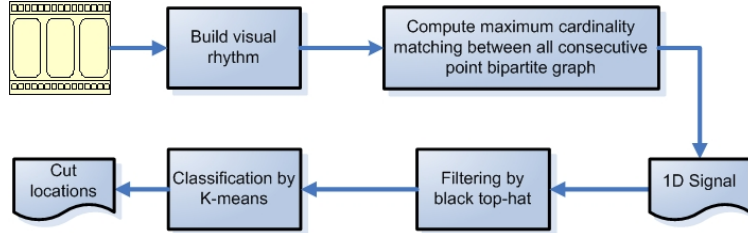


Figure 2. Workflow for transition detection

One should notice that Δ may be either specified or an adaptive threshold can be used. As it is presented in this work, adaptive thresholding is performed by a k-means clustering algorithm in order to identify the cut location.

3 Methodology for transition detection

As described before, the main goal of the transition detection problem is to identify sudden changes on a video sequence, indicating the occurrence of different visual events, as described in Fig. 2.

As it can be seen by the presented workflow, the first step of the process is the computation of the visual rhythm. The choice of an appropriate feature vector that enhances the performance of matching algorithm is not a trivial task. Therefore, empirical studies are the best way to get insights of which feature should be used for each case. Then, a point similarity graph is constructed based on a given threshold value for each pair of frames and the cardinality of each generated graph is computed and stored in the form of a 1-dimensional signal.

The algorithm is based on the hypothesis that video transitions will produce sudden changes in the cardinality of the point similarity graphs. In order to identify these transitions on the cardinality sequence, a simple morphological black top-hat filter is used find local minima with high dynamics [1]. The main difference between our approach and the method proposed by [1] is the replacement of the dissimilarity distance. While [1] uses the size of the longest common subsequence, we consider the maximum cardinality matching.

After the filtering step, a transition is defined as a local maximum that is larger than an specified threshold value. Since this threshold value may depend on video content, a more robust approach is to use the K-means clustering algorithm to classify the cardinality sequence in the “cut” and “non-cut” clusters. K-means starts with 2 centroids defined by the min and max values of the 1-dimensional signal of the cardinalities. After the clustering procedure, the cluster with the highest values corresponds to cut transitions. The Algorithm 1 illustrates our method for cut detection.

Algorithm 1 Cut detection

Require: Video sequence

Video (V_M^T)

Visual Rhythm (VR)

Threshold value (δ)

{ M = size of the video}

{ N = height of the visual rhythm}

1: count = 0; k = 1;

2: signal = \emptyset

3: **while** ($k \leq M - 1$) **do**

4: “Construct G^δ to f_{t_k} and $f_{t_{k+1}}$ ”;

5: “Calculate \overline{M}_k^δ for G^δ ”

6: “Insert $|\overline{M}_k^\delta|$ to signal”

7: **end while**

8: “Apply black top-hat to signal”

9: $G1 = \min\{\text{filtered signal}\}$

10: $G2 = \max\{\text{filtered signal}\}$

11: $\langle G1, G2 \rangle = \text{KMeans}(\text{filtered signal})$

12: **return** G2

4 Experiments

In the experiment that follows, we used the same dataset considered in [11]. The video clips represent a variety of different video genres. According to [11], its method has a better performance when compared to the pixel based and the method proposed by [8]. Table 1 presents a comparison of three quality measures (recall, precision and F1 score) for the following methods: (i) feature based with automatic threshold selection [11]; (ii) histogram based method [8]; (iii) LCS method [1]; (iv) pixel based method; and (v) the proposed method with clustering by k-means. As it can be seen in Table 1, the performance results of our method are very similar to the feature based method. However, in seven cases, the F1 score of our method is greater than or equal to the feature based. The performance rates of our method are better when compared to the LCS approach.

The algorithms presented in Table 1 covers different approaches to the transition detection problem.

While the feature based method with automatic thresholding presents good results, the histogram based

Data Source	Our proposal			LCS method			Feature tracking method			Pixel Based method with localization			Histogram method Cut Det (MOCA)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
A lisa	1	1	1	1	0.857	0.923	1	1	1	1	1	1	1	1	1
B jamie	0.500	1	0.667	0.096	1.000	0.176	1	1	1	0.825	0.825	0.825	1	0.375	0.545
C psycho	0.662	0.907	0.766	0.635	0.870	0.734	0.595	0.870	0.707	0.764	0.778	0.771	0.936	0.536	0.682
D sexinthecity	1	1	1	1.000	0.971	0.985	1	1	1	1	1	1	1	0.941	0.969
E highlander	0.828	0.857	0.842	0.676	0.821	0.742	0.938	1	0.968	0.867	0.867	0.867	0.955	0.7	0.808
F comercial2	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
G commercial	0.950	1	0.974	1	0.842	0.914	0.810	0.944	0.872	0.708	0.994	0.809	1	0.667	0.800
H abstract	0.949	0.974	0.961	0.943	0.868	0.904	0.895	0.895	0.895	0.927	1	0.962	0.971	0.895	0.932
I news	1	1	1	0.667	0.500	0.571	1	1	1	1	1	1	1	0.5	0.667
J law	0.683	0.869	0.765	0.639	0.885	0.742	0.497	0.897	0.637	0.623	0.54	0.591	0.85	0.395	0.54
Weighted mean	0.804	0.929	0.857	0.756	0.877	0.794	0.754	0.927	0.820	0.805	0.820	0.811	0.941	0.644	0.750

Table 1. Experimental results (adapted from [11])

method, on the other hand, is a simple and well-known technique that provides good results. The principle of our method is closely related to the LCS method principle, in which the main difference relies on the replacement of the LCS computation by the maximum cardinality of the bipartite graph matching.

5 Conclusion and further works

In this work, we used the maximum cardinality of the bipartite graph matching between two frames as a dissimilarity distance in order to identify cut locations. The main contribution of our work is the application of a simple distance to solve a problem of video segmentation.

Many others in literature achieve same precision and recall rates using more expensive (time-consuming) approaches. Rotation and translation invariance is another important feature of our approach without increasing computational time (it is a natural property of graph matching). According to experimental results, the performance of our method is similar to the method proposed by [11] with lower computational cost. When compared to the method proposed by [1], our method presented higher recall and precision rates.

Nonetheless, the algorithm presented a high occurrence of false positives, which might be related to the use of the visual rhythm to compute the point similarity graph, since it reduces frame information. As a future work, we plan to apply our approach directly to video frames (instead of using a visual rhythm) in order to cope with these difficulties

Acknowledgments

The authors are grateful to PUC Minas (Pontifícia Universidade Católica de Minas Gerais) and CT-Info/MCT/CNPq (Project 551005/2007-6) for the financial support of this work.

References

- [1] F. N. Bezerra and N. J. Leite. Using string matching to detect video transitions. *Pattern Anal. Appl.*, 10(1):45–54, 2007.
- [2] M. G. Chung, J. Lee, H. Kim, S. M.-H. Song, and W. M. Kim. Automatic video segmentation based on spatio-temporal features. *Korea Telecom Journal*, 4(1):4–14, 1999.
- [3] S. J. F. Guimarães, M. Couprie, N. J. Leite, and A. A. Araújo. A method for cut detection based on visual rhythm. In *Proc. of the XIV Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI*, pages 297–304, Brazil, Oct 2001. IEEE Computer Society Press, ISBN 0769513301.
- [4] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *SPIE Image and Video Processing VII*, Jan 1999.
- [5] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, 1999.
- [6] C. W. Ngo, T. C. Pong, and R. T. Chin. Detection of gradual transitions through temporal slice analysis. In *Proc. of the IEEE CVPR*, pages 36–41, 1999.
- [7] Z. K. G. d. Patrocínio Júnior, S. J. F. Guimarães, and H. B. de Paula. Bipartite graph matching for video clip localization. In A. X. Falcão and H. C. V. Lopes, editors, *Proceedings*. IEEE Computer Society, Oct. 7–10, 2007 2007.
- [8] S. Pfeiffer, R. Lienhart, G. Kuhne, and W. Effelsberg. The moca project - movie content analysis research at the university of mannheim. In *GI Jahrestagung*, pages 329–338, 1998.
- [9] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata. Videomap and videospaceicon: Tools for anatomizing video content. In *ACM Interchi*, pages 131–136, 1993.
- [10] Y. Wang, Z. Liu, and J.-C. Huang. Multimedia content analysis. *IEEE Signal Processing Magazine*, pages 12–36, 2000.
- [11] A. Whitehead, P. Bose, and R. Laganier. Feature based cut detection with automatic threshold selection. In *CIVR04*, pages 410–418, 2004.
- [12] R. Zabih, J. Miller, and K. Mai. Feature-based algorithms for detecting and classifying scene breaks. In *ACM ICMCS*, USA, Nov 1995.