Bipartite graph matching for video clip localization

Zenilton Kleber G. do Patrocínio Jr. Silvio Jamil F. Guimarães
Hugo Bastos de Paula
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)
Rua Walter Ianni, 255 - São Gabriel - 31980-110 - Belo Horizonte - MG - Brazil
{zenilton,sjamil,hugo}@pucminas.br

Abstract

Video clip localization consists in identifying real positions of a specific video clip in a video stream. To cope with this problem, we propose a new approach considering the maximum cardinality matching of a bipartite graph to measure video clip similarity with a target video stream which has not been preprocessed. We show that our approach locates edited video clips, but it does not deal with insertion and removal of frames/shots, allowing only changes in the temporal order of frames/shots. All experiments performed in this work have achieved 100% of precision for two different video datasets. And according to those experiments, our method can achieve a global recall rate of 90%.

1. Introduction

Traditionally, visual information has been analogically stored and manually indexed. Due to advances in multimedia technology, techniques to video retrieval are increasing. Unfortunately, the recall and precision of these systems depend on the similarity measure that are used to retrieve information. Nowadays, due to improvements on digitalization and compression technologies, database systems are used to store images and videos, together with their metadata and associated taxonomy. Thus, there is an increasing search for efficient systems to process and index image, audio and video information, mainly for the purposes of information retrieval.

The task of automatic segmentation, indexing, and retrieval of large amount of video data has important applications in archive management, entertainment, media production, rights control, surveillance, and many more. The complex task of video segmenting and indexing faces the challenge of coping with the exponential growth of the Internet, that has resulted in a massive publication and sharing of video content and an increase in the number of duplicated documents; and the distribution across communication channels, like TV, resulting in thousands of hours of streaming broadcast media. According to (6; 7; 8), one important application of video content management is broadcast monitoring for the purpose of market analysis. The video clip localization, as it will be referred during this paper, has arisen in the domain of broadcast television, and consists of identifying the real locations of a specific video clip in a target video stream (see Fig. 1). The main issues that must be considered during video clip localization are: (i) the definition of the dissimilarity measures of video clips; (ii) the processing time of the algorithms due to the huge amount of information that must be analyzed; (iii) the insertion of intentional and non-intentional distortions; and (iv) different frame rates. The selection of the feature used to compute dissimilarity measure has an important role in content-based image retrieval and has been largely explored (20). (5) showed that the performance of the features is task dependent, and that it is hard to select the best feature for an specific task without empirical studies. Low-complexity features and matching algorithms can work together to increase matching performance. This work uses a low complexity feature to test a novel matching procedure. Feature selection will not be addressed in this paper.

Current methods for solving the video retrieval/localization problem can be grouped in two main approaches: (i) computation of video signatures after temporal video seg-

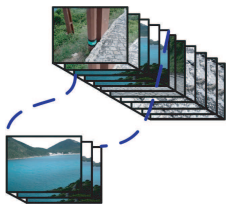

Figure 1. Problem of identifying the real position of a specific video clip in a target video stream.

	Sliding window (3)	Temporal order (11; 21)	Vstring edit (1)	Multi-level (14)	Graph approach (19)	BMH (9)	Our proposed method
Shot/Frame Matching	shot	shot	shot	shot	shot	frame	frame
Temporal order	no	yes	yes	yes	yes	yes	possible
Clip filtering	no	no	no	no	yes	no	no
Online Clip Segment.	yes	no	no	no	yes	no	no
Preprocessing	yes	yes	yes	yes	yes	no	no
Video edition	no	no	no	no	no	no	partially

Table 1. Comparison of some approaches for video clip localization (adapted from (19))

mentation, as described in (7; 12; 15); and (ii) use of matching algorithms after transformation of the video frame content into a feature vector, as described in (1; 13; 9). When video signatures are used, methods for temporal video segmentation must be applied before signature calculation (2). Although temporal video segmentation is a widely studied problem, it represents an important issue that has to be considered, as it increases complexity of the algorithms and affects matching performance. For methods based on string matching algorithms, the efficiency of the these algorithms must be taken into account, when compared to image/video identification algorithms. (1) and (13) successfully applied the longest common substring (LCS) algorithm to deal with the problem. However, it requires a $O(mn)$ space and time cost, in which m and n represent the size of the query and target video clips, respectively. In (9), it is proposed a modified version of the fastest algorithm for exact string matching, the Boyer-Moore-Horspool (BMH) (10; 16), to deal with the problem of video location and counting.

In the present paper, we propose a new approach to cope with the problem of video clip localization using the maximum cardinality matching of a bipartite graph. For a set of frames from a query video clip and from a target video a graph is constructed based on a similarity measure between each pair of frames (illustrated in Fig. 2(a)). The size of the maximum cardinality matching of the graph defines a video similarity measure that is used for video identification. Table 1 presents a comparison between some approaches found in the literature. The first difference between our proposed approach and the others is associated with the matching used to establish the video similarity. Most of the works consider that the target video has been preprocessed and online/offline segmented into video clips which are used by the search procedure, while ours can be applied directly to a target video stream without any preprocessing since it uses frame-based similarity measures. With the exponential growth of the Internet, the storage of segmented videos may become an intractable problem. Our approach allows us to perform video localization over a streaming media downloaded directly from the Internet, while the others need to download, segment and store segmented video clips before starting video clip localization.

Moreover, our approach can be applied without considering temporal order constraints, which allows us to locate the position of the query video even if the video has been edited (see Fig. 2(b)). Current version of our algorithm does not deal with insertion and removal of frames/shots, but it allows changes in temporal order of query video clip frames/shots. Clip editing and reordering has become a desired feature on the new context of online video delivery. As mentioned in (18), users expect to be able to manipulate video content based on choices such as desired portions of video, ordering and "crop/stitch" of clips. New coding schemes that consider this novel scenario have been included in most recent standards such as MPEG-7 and MPEG-21 (22). However, using dynamic programming and temporal order similarity, our approach can be applied to the traditional (exact) video clip localization problem. Nevertheless, since our approach is based on frame similarity measures, it may present efficiency problem. This issue has been addressed by employing a *shift strategy* based on the size of the maximum cardinality matching.

This paper is organized as follows. In Sec. 2, the problem of video clip localization is described, together with some formal definitions of the field. In Sec. 3, we present a methodology to identify the location of a video clip using bipartite graph matching. In Sec. 4, we discuss about the experiments and the setting of algorithm parameters. Finally, in Sec. 5, we give some conclusions and future works.

2. Problem definition to video clip localization

Let $\mathbb{A} \subset \mathbb{N}^2$, $\mathbb{A} = \{0, \ldots, \mathrm{H}-1\} \times \{0, \ldots, \mathrm{W}-1\}$, where H and W are the width and height of each frame, respectively, and, $\mathbb{T} \subset \mathbb{N}$, $\mathbb{T} = \{0, \ldots, N-1\}$, 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*

$$\mathrm{V}_N = (f)_{t \in \mathbb{T}} \tag{1}$$

(a) query video without edition

(b) edited (frame-reordered) query video

Figure 2. Frame similarity graph

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

Definition 2.3 (Video clip) *Let V_N be a video. A j-sized video clip (or sequence) $\mathrm{S}_{k,j}$ is a temporally ordered set of frames from V_N which starts at frame k and it can be described by*

$$\mathrm{S}_{k,j} = (f_t | f_t \in \mathrm{V}_N)_{t \in [k, k+j-1]}. \quad (2)$$

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

Definition 2.4 (Frame similarity) *Let f_{t_1} and f_{t_2} be two video frames at location t_1 and t_2, respectively. Two frames are similar if a distance measure $\mathcal{D}(f_{t_1}, f_{t_2})$ between them is smaller than a specified threshold (δ). The frame similarity is defined as*

$$FS(f_{t_1}, f_{t_2}, \delta) \begin{cases} 1, & \text{if } \mathcal{D}(f_{t_1}, f_{t_2}) \leq \delta \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

There are several choices for $\mathcal{D}(f_{t_1}, f_{t_2})$, i.e., the distance measure between two frames, e.g. histogram/frame difference, histogram intersection, difference of histograms means, and others.

After selecting one, it is possible to construct a frame similarity graph based on a query video V_M^Q and M-sized video clip of target video $S_{k,M}^T$ as follows.

Definition 2.5 (Frame similarity graph – G_k^δ) *Let V_M^Q and V_N^T be a query video with M frames and a target video with N frames, respectively, and let $\mathrm{S}_{k,M}^T$ be a M-sized video clip which starts at frame k of target video. A frame similarity graph $\mathrm{G}_k^\delta = (N^Q \cup N_k^T, \mathrm{E}_k^\delta)$ is a bipartite graph. Each node $v_{t_1}^Q \in N^Q$ represents a frame $f_{t_1}^Q \in \mathrm{V}_M^Q$ and each node $v_{t_2}^T \in N_k^T$ represents a frame $f_{k+t_2}^T \in S_{k,M}^T$. There is an edge $e \in \mathrm{E}_k^\delta$ between $v_{t_1}^Q$ and $v_{t_2}^T$ if frame similarity of associated frames is equal to 1, i.e.,*

$$\mathrm{E}_k^\delta = \{ (v_{t_1}^Q, v_{t_2}^T) \mid v_{t_1}^Q \in N^Q, v_{t_2}^T \in N_k^T, FS(f_{t_1}^Q, f_{k+t_2}^T, \delta) = 1 \} \quad (4)$$

As illustrated in Fig. 2, we match the query video to a video clip of the target video stream with the same size (number of frames), although it is possible to relax this constraint in order to allow video clip editions that insert and/or remove frames/shots. In this paper, we focus on video clip localization problem without any changes in the video content (only in its temporal order). To do so, we define *matching* and *maximum cardinality matching* as follows.

Definition 2.6 (Matching – M_k^δ) *Let $\mathrm{G}_k^\delta = (N^Q \cup N_k^T, \mathrm{E}_k^\delta)$ be a frame similarity graph. A subset $\mathrm{M}_k^\delta \subseteq E_k^\delta$ is a match if any two egdes in M_k^δ are not adjacent.*

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

Finally, video clip localization problem can be defined.

Definition 2.8 (Video clip localization – VCL) *The video clip localization (VCL) problem corresponds to the identification of a query video V_M^Q that belongs to a target video V_N^T if there is a video clip $\mathrm{S}_{k,M}^T$ of V_N^T that matches with V_M^Q according to the frame similarity. Thus, this problem can be defined by*

$$VCL(\mathrm{V}_M^Q, \mathrm{V}_N^T, \delta) = \{ k \in \mathbb{T} \mid |\overline{\mathrm{M}_k^\delta}| = M \} \quad (5)$$

where $\overline{\mathrm{M}_k^\delta}$ is the maximum cardinality matching of a frame similarity graph G_k^δ which is generated using the query video V_M^Q, a video clip $\mathrm{S}_{k,M}^T$ that starts at frame k and specified threshold δ.

3. Methodology for the video clip localization problem

As described before, the main goal of the video clip localization problem is to identify occurrences of a query video in a video stream, see Fig. 3. One of the key steps of the process is feature extraction. Choosing an appropriate feature that enhances performance of a 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.

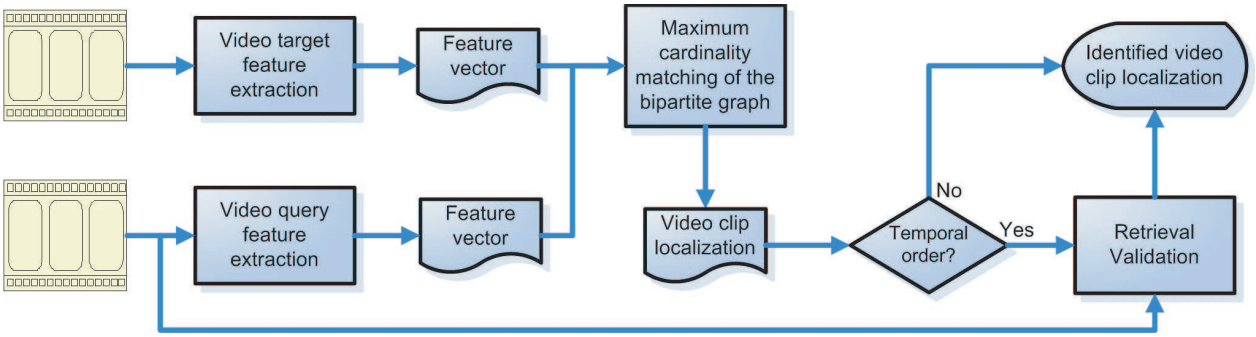

Figure 3. Workflow for video clip localization

3.1. Search procedure

Algorithm 1 presents our search procedure. It scans over target video, looking for a video clip that matches the query video, i.e., one that generates a frame similarity graph (*line 3*) which has a maximum cardinality matching with size equal to query video size (*lines 4-5*).

Table 2 shows the size of the maximum cardinality matching and the shift value for a video clip localization in which the target video is represented by feature values $(1, 5, 6, 2, 4, 2, 1, 3, 5, 1, 2, 3, 7, 6, 1)$ and the query video by $(1, 2, 3)$. The query video appears at two distinct positions and the search procedure has identified both. First occurrence has a temporal order that is different from the query video order, while the other is an exact match.

It also important to describe the *shift strategy* adopted (at *line 9* and *line 11* of Algorithm 1). After locating a match, the procedure ensures a jump that is equal to the query video size (*line 9*) since one should not expect to find the query video inside itself. This not only contributes to accelerate the search but it also helps reducing the number of false positives, i.e., the number of video occurrences that do not represent a correct identification. In fact, the size of the maximum cardinality matching could be almost equal to the query video size for some iterations close to the *hit* positions, depending on query video content and size. That could slow down the search. Using a shift value equals to the query video size does not improve performance before a *hit* position but it prevents a performance reduction after the *hit* position has been found.

In case of a mismatch, the shift value is set to the difference between the query video size and the size of the maximum cardinality matching, i.e., the number of unmatched frames (*line 11*). In spite of being a conservative approach, this setting allows our search procedure to perform better than the naïve (brute force) algorithm and it could result in a great performance improvement depending on query video content and size, e.g., the search procedure would be faster for query videos that are more dissimilar from target video.

It is also important that the search procedure does not miss a *hit* position. Adjusting the shift value to the number of unmatched frames avoids that by using a conservative approach which assumes that all mismatches occurred in the beginning of the video clip $\mathrm{S}_{k,M}^{T}$ of the target video. So, it is necessary to shift the video clip of the target video at least the same number of unmatched frames in order to be feasible to find a new *hit* position.

Generation of frame similarity graph (*line 3*) and calculation of the maximum cardinality matching (*line 4*) are the most time consuming steps of Algorithm 1. Graph generation needs $O(M^2)$ operations, in which M represents the query video size, and total time spent on graph generation is $O(NM^2)$, if shift value set to the its worst possible value, i.e., if it is equal to 1.

Algorithm 1 Search procedure

Require: Video sequences
Target video (V_N^T)
Query video (V_M^Q)
Threshold value (δ)
{M = size of the query video}
{N = size of the target video}
{pos = containing query video positions at the target}

1: count = 0; k = 0;
2: **while** (k $\leq N - M + 1$) **do**
3: "Construct G_k^δ";
4: "Calculate $\overline{\mathrm{M}_k^\delta}$ for G_k^δ"
5: **if** $|\overline{\mathrm{M}_k^\delta}| = M$ **then**
6: "Query video was found at position k"
7: pos[count] = k
8: count = count + 1;
9: k += $|\overline{\mathrm{M}_k^\delta}|$
10: **else**
11: k += $M - |\overline{\mathrm{M}_k^\delta}|$
12: **end if**
13: **end while**
14: **return** pos

Video Information																Maximum Card. Matching Size	Shift Value	Iteration Number
Target video	1	5	6	2	4	2	1	3	5	1	2	3	7	6	1	-	-	-
Query video	1	2	3													1	2	1
			1	2	3											1	2	2
					1	2	3									2	1	3
						1	**2**	**3**								**3**	3	4
									1	2	3					2	1	5
										1	**2**	**3**				**3**	3	6
													1	2	3	1	-	7
														Average		1.86	2.00	-

Table 2. Example of video clip matching

Solving the maximum cardinality matching on a bipartite graph could done with $O(E\sqrt{V})$ operations (17), in which V and E represent the number of nodes and edges, respectively. The number of nodes is always equal to $2M$, while the number of edges depends on frame similarity measures and threshold. It could be close to zero, but it also could be equal to M^2 in the worst case scenario.

One should notice that at least M edges are needed in order to find a *hit* position, i.e., the size of the maximum cardinality matching has to be equal to M. And, for practical reasons, one should consider the number of edges to be at least $O(M)$ in the iteration that locates a *hit* position. So, the maximum cardinality matching should need at least $O(M\sqrt{M})$ operations, but it could take $O(M^2\sqrt{M})$ in the worst case scenario.

Assuming that all query video frames are similar to all target video frames is quite unrealistic since frame similarity measure and threshold should reduce that number. Moreover, this worst case scenario always leads to the optimal shift value, i.e., a shift value equals to the query video size because maximum cardinality matching for a complete bipartite graph $K_{M,M}$ has size equals to M. The search algorithm runs faster when optimal shift value is used (only $O(N/M)$ positions need to be tested), and total time spent on maximum cardinality matching calculation is $O(NM\sqrt{M})$.

Thus, our search procedure has a time complexity of $O(NM^2)$ since it is dominated by total time spent in the graph generation step.

3.2. Retrieval validation

After query video position candidates are selected by the search algorithm, the results must be validated to ensure that no false positives are considered when there is some assumptions, like temporal order. To verify this assumption, we can use the dynamic programming (DP), as proposed by (19). We define a temporal order similarity measure as follows.

Definition 3.1 (Temporal order similarity – TS_δ) *Let i be i-th frame of the query video V_M^Q, i.e., $i = f_i \in \mathrm{V}_M^Q$, and j be j-th frame of the video clip $\mathrm{S}_{k,M}^T$ of the target video, i.e., $j = f_j \in \mathrm{S}_{k,M}^T$. The temporal order similarity $\mathrm{TS}_\delta(\mathrm{V}_M^Q, \mathrm{S}_{k,M}^T)$ between the query video and the video clip of the target video is equal to $\mathrm{T}_\delta(M, M)$ which is calculated using DP as follows*

$$\mathrm{T}_\delta(i,j) = \begin{cases} \emptyset & \text{if } i = 0 \text{ or } j = 0 \\ \mathrm{T}_\delta(i-1, j-1) + 1 & \text{if } FS(i,j,\delta) = 1 \\ \max\{\mathrm{T}_\delta(i, j-1), & \\ \qquad \mathrm{T}_\delta(i-1, j)\} & \text{otherwise} \end{cases} \quad (6)$$

where δ is a specified frame similarity threshold.

Using the temporal order similarity measure TS_δ, we can validate query video position candidates and eliminate false positives by ensuring that temporal order similarity between the query video and the video clip of the target video is greater than a threshold, i.e., $\mathrm{TS}_\delta(\mathrm{V}_M^Q, \mathrm{S}_{k,M}^T) \geq \Delta$, in which Δ represents the minimum number of similar frames in correct temporal order that should be found in order to accept a query video position candidate. No temporal order changes are allowed, if Δ is set to the query video size, i.e., $\Delta = M$.

4. Experiments

In this section, we present two experiments with their respective analysis. For each experiment, we consider two different datasets. The first one consists of Broadcast TV commercials, recorded directly and continuously from a brazilian cable TV channel, while the second was constructed from various sources, including the Internet, with mixed different qualities and compression standards, followed by a edition of these videos using cut transition. Table 3 shows the length of these two datasets.

Experiments of video clip localization were performed over the datasets. The experiments searched for 20 occurrences of video clips for the TV broadcast dataset and 47

Video dataset	Time	Frame rate
TV Broadcast Commercials	2h 52m 31s	30 fps
Internet Retrieved Video	1h 36m 44s	30 fps
Total	4h 29m 15s	-

Table 3. Target video corpora

occurrences of video clips for the Internet retrieved dataset. The similarity graph (G_k^δ) was constructed using the difference of frame histogram means as distance measure; and the similarity threshold δ was set to 0, 1, 2 and 3. Figure 4 shows two sample runs of the localization algorithm.

The curves in Figure 4 represent the size of the maximum cardinality matching found by our algorithm at a certain position. In the upper plot, the query video appears at three different positions of the target video. In the lower plot the query video occurs only once. A query video clip occurrence is stated when the size of maximum cardinality matching is equal to the length of the query video. As described in the previous section (see Table 2), at every iteration of the localization procedure, the search is shifted by a number of frames that depends on the length of the query video and the size of the current maximum cardinality matching. An optimal shift should be equal to the query video length. Figure 4 also shows the average ratio between shift and query video length. The first run (upper) has achieved 63.8% of shift length and the second run (lower) has showed a 42.4% of shift length, compared to its respective query length. These results will be discussed later on this article.

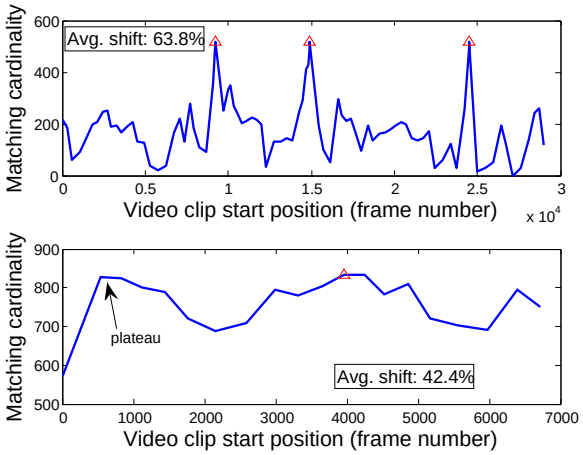

Figure 4. Localization procedure. The line represents the size of maximum cardinality matching at different positions of a video stream. The points of maximum (triangle) correspond to a full matching, which identifies an occurrence of the query video. The average shift is shown as a percentage of the query video length.

Average Recall (%)				
Video dataset	**Threshold value (δ)**			
	0	**1**	**2**	**3**
TV Broadcast Commercials	20	65	75	80
Internet Retrieved Video	0	16.6	43	94.8
Global average	6	31	52.6	90.4

Table 4. Recall percentage for different similarity threshold values (δ).

4.1. Precision-Recall Analysis

In order to evaluate the results, it is necessary to define some measures. We denote by $\#Occurrences$ the number of query video occurrences, by $\#Video\ clip\ identified$ the number of query video occurrences that are properly identified and by $\#Falses$ the number of video occurrences that do not represent a correct identification. Based on these values, we consider the following quality measures.

Definition 4.1 (Recall and precision rates) *The recall rate represents the ratio of correct and the precision value relates correct to false detections. These measures are given by*

$$\alpha = \frac{\#Video\ clip\ identified}{\#Occurrences} \quad (recall) \qquad (7)$$

$$\mathcal{P} = \frac{\#Video\ clip\ identified}{\#Falses + \#Video\ clip\ identified} \quad (precision) \qquad (8)$$

Precision-Recall (PR) curves give a more informative picture of the algorithm performance, since they group information about hits, miss, false positives and false negatives (4). An optimal algorithm should have a precision-recall value of (1,1) (which means 100% of recall with 100% of precision).

All experiments performed in this work have achieved 100% of precision, for every δ in both datasets. One of the parameters of the algorithm that contributes to this result is the size of the maximum cardinality matching. The algorithm only considers a video clip position as a positive localization if the size of the maximum cardinality matching is equal to the query video length. In other words, only if all frames of the query video have a match, a hit is found. Using the size of the maximum cardinality matching prevents the algorithm from finding query videos that have additional frames/shots, or target videos that suppress parts

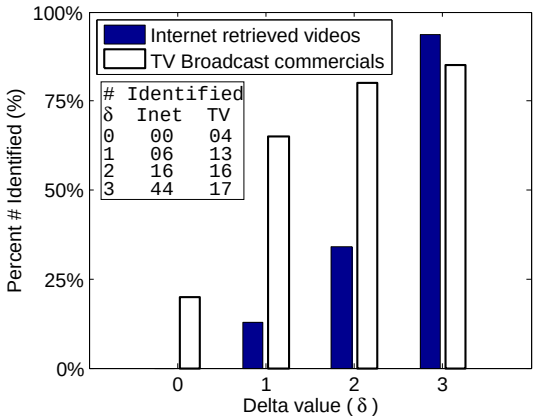

Figure 5. Recall percentage for different similarity threshold values (δ).

(frames/shots) of the query video. A relaxation of this constraint is possible, but it may imply in a rise of false positive number.

Another parameter that contributes to high precision detection is the similarity threshold value (δ), that is kept very low in the presented experiments (up to 3). Low threshold values prevent the algorithm from achieving high recall rates for videos with degradation or different resolutions. Again, raising this value may increase recall rates, but it also may increase the number of false positives. Table 4 presents the average recall percentage over all query videos.

Figure 5 shows the results of Table 4. It highlights that our localization algorithm has had a significantly worse performance ($\approx$ 30% and lower) on the Internet dataset for low values of δ. Differences of coding schemes and frame resolutions among the videos retrieved from the Internet might be one of the reasons for that difference in performance. Nevertheless, provided that TV Broadcast Commercials were recorded under the same circumstances, a similarity threshold value (δ) equals to 1 has been enough to achieve 65% of recall rate.

4.2. Performance Issues

Our work can be directly compared to the methods developed by Adjeroh et al. (1) and Kim et al. (13). While these works consider edit distance to compute the similarity between videos in which insertion, remotion and substitution are permitted, our method works very well when the frame rate is constant and no insertion or remotion is applied. However, our approach allows changes in the order of frames. It also avoid preprocessing of target video, e.g.

Average (Std. Dev.) Shift Value (%)				
Video dataset	**Threshold value (δ)**			
	0	**1**	**2**	**3**
TV Commercials	54 (12)	50 (13)	46 (14)	43 (15)
Internet Video	61 (15)	52 (14)	47 (14)	47 (15)
Global average	58	51	46	45

Table 5. Average (std. deviation) shift value percentage for different similarity threshold values (δ).

clip segmentation, working directly over the frames, thus saving CPU time.

The performance of the algorithm is directly related to the shift that is applied to the target video after a matching procedure. Once the size of the maximum cardinality matching is calculated, the conservative approach used in this work defines the shift value as the number of unmatched frames. This approach assumes that all mismatches occurred in the beginning of the graph. In other words, the algorithm considers that all matches might be used in the next iteration, preventing the algorithm to shift at larger steps.

Table 5 shows the average shift value of the performed experiments. A number of 100% means that the shift value is equal to the query video length (optimal situation). It can be seen that, at lower values of δ the average shift value is higher. This effect is expected since a lower value of δ increases the number of mismatched frames.

The standard deviation of shift value was stable around 14% for every dataset and δ. The global results show mean shift values around 40-50% which means that there is much room for improvement in the algorithm performance regarding to that parameter.

5. Conclusions

In this work, we proposed the utilization of the maximum cardinality matching to deal with the problem of video clip localization in which target video stream is not preprocessed, i.e., it is not segmented into video clips.

Our approach can also be applied without considering temporal order constraints, which allows us to locate the query video position even if the video has been edited. Current version of our algorithm does not deal with insertion and removal of frames/shots, but it allows changes in temporal order of query video clip frames/shots. Exploring its capacity to deal with other editing operations can be seen as a future work.

Since our localization algorithm is based on frame similarity measure, it may present efficiency problems. This issue has been addressed by employing a *shift strategy* based

on the size of the maximum cardinality matching. Another future research line may propose and evaluate other shift strategies.

Finally, all experiments performed in this work have achieved 100% of precision for both datasets. And according to those experiments, our method can achieved a global recall rate of approximately 90% after adjusting the frame similarity threshold. Raising the frame similarity threshold value may increase recall rates, but it also may increase the number of false positives. Future works may also try to increase recall rates without increasing the number of false positives.

6. Acknowledgments

The authors wish to thank the anonymous referees for their very valuable comments. The authors are also grateful to PUC Minas and to Instituto de Informática.

References

[1] D. A. Adjeroh, M.-C. Lee, and I. King. A distance measure for video sequences. *Computer Vision and Image Understanding*, 75(1-2):25–45, 1999.

[2] A. D. Bimbo. *Visual information retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[3] L. Chen and T.-S. Chua. A match and tiling approach to content-based video retrieval. In *ICME*. IEEE Computer Society, 2001.

[4] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proc. of the 23rd International Conference on Machine Learning*, pages on–line, Pittsburgh, PA, June 2006.

[5] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval – a quantitative comparison. In *DAGM 2004, Pattern Recognition, 26th DAGM Symposium, Lecture Notes in Computer Science*, pages 228–236, Tübingen, Germany, September 2004.

[6] N. Diakopoulos and S. Volmer. Temporally tolerant video matching. In *Proc. of the ACM SIGIR Workshop on Multimedia Information Retrieval*, Toronto, Canada, August 2003.

[7] J. Gauch and A. Shivadas. Identification of new commercials using repeated video sequence detection. In *International Conference on Image Processing*, pages III: 1252–1255, 2005.

[8] J. M. Gauch and A. Shivadas. Finding and identifying unknown commercials using repeated video sequence detection. *Computer Vision and Image Understanding: CVIU*, 103(-):80–88, / 2006.

[9] S. J. F. Guimarães, R. Kelly, and A. Torres. Counting of video clip repetitions using a modified bmh algorithm: Preliminary results. In *Proc. of the IEEE ICME*, pages 1065–1068, Toronto, Canada, July 2006.

[10] R. N. Horspool. Practical fast searching in strings. *Softw., Pract. Exper.*, 10(6):501–506, 1980.

[11] A. K. Jain, A. Vailaya, and W. Xiong. Query by video clip. *Multimedia Syst.*, 7(5):369–384, 1999.

[12] A. Joly, C. Frelicot, and O. Buisson. Content-based video copy detection in large databases: A local fingerprints statistical similarity search approach. In *International Conference on Image Processing*, pages I: 505–508, 2005.

[13] Y. Kim and T. Chua. Retrieval of news video using video sequence matching. In *MMM*, pages 68–75, 2005.

[14] R. Lienhart, W. Effelsberg, and R. Jain. Visualgrep: A systematic method to compare and retrieve video sequences. *Multimedia Tools Appl.*, 10(1):47–72, 1999.

[15] X. Naturel and P. Gros. A fast shot matching strategy for detecting duplicate sequences in a television stream. In *Proceedings of the 2nd ACM SIGMOD International Workshop on Computer Vision meets DataBases*, 2005.

[16] G. Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, 2001.

[17] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[18] J. S. Pedro, N. Denis, and S. Domínguez. Video retrieval using an edl-based timeline. In J. S. Marques, N. P. de la Blanca, and P. Pina, editors, *IbPRIA (1)*, volume 3522 of *Lecture Notes in Computer Science*, pages 401–408. Springer, 2005.

[19] Y. Peng and C.-W. Ngo. Clip-based similarity measure for query-dependent clip retrieval and video summarization. *IEEE Trans. Circuits Syst. Video Techn.*, 16(5):612–627, 2006.

[20] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding: CVIU*, 84(1):25–43, October 2001.

[21] Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge. A framework for measuring video similarity and its application to video query by example. In *ICIP (2)*, pages 106–110, 1999.

[22] B. L. Tseng, C.-Y. Lin, and J. R. Smith. Using MPEG-7 and MPEG-21 for personalizing video. *IEEE MultiMedia*, 11(1):42–53, 2004.