A Comparison of Distance Measures for Clustering Video Sequences

Werner Bailer
Institute of Information Systems & Information Management
JOANNEUM RESEARCH Forschungsgesellschaft mbH
Steyrergasse 17, 8010 Graz, Austria
werner.bailer@joanneum.at

Abstract

Matching video segments in order to detect their similarity is a necessary task in retrieval and summarization applications. In order to determine nearly identical content, such as repeated takes of the same scene, very precise matching of sequences of features extracted from the video segments needs to be performed. In this paper we compare the performance of three distance measures for the task of clustering multiple takes of the same scene: Dynamic Time Warping (DTW) and two variants of Longest Common Subsequence (LCSS). We also evaluate the influence of the quality of the input segmentation on the performance of the algorithms.

1. Introduction

A collection of video material often has a high degree of redundancy, not only due to the reuse of identical video segments, but also because there are segments that show nearly identical content. Examples are several takes of a scene in rushes material or an event recorded from several very similarly positioned cameras, which is a typical case in news covered by different broadcasters. These video segments do not only share similar static properties (e.g. color distribution in a frame), but are also similar over time (e.g. camera motion, movement of actors and objects). But they are not identical: objects are at slightly different positions and the temporal alignment of the segment may vary, i.e. there may be omissions and insertions.

In this paper we present an overview of algorithms for measuring similarity of video segments (Section 2). We select two suitable classes of measures for matching sequences of feature vectors, Dynamic Time Warping (DTW) and Longest Common Subsequence (LCSS), and discuss variants of these approaches in more detail in Section 3. In Section 4 we compare the performance of these approaches on the TRECVID 2007 BBC rushes data set [6] and evaluate the robustness of the approaches against errors in the underlying temporal video segmentation. Section 5 concludes the discussion.

2 State of the art

There are many approaches for similarity matching of video clips, many of them applying still image features for clustering key frames of video sequences. However, similarity matching in these approaches is often quite coarse. The approaches are optimized toward compact feature descriptions and scalability rather than precise matching of the content of a sequence.

The most prominent applications for near-duplicate detection are finding illegal copies of video content or the identification of known unwanted content in public access video databases. These applications are based on the following assumptions: (i) the actual content of the videos to be matched is identical, (ii) partial matches need to be identified and (iii) the algorithm needs to be robust against a number of distortions, such as changes of sampling parameters, noise, encoding artifacts, cropping, change of aspect ratio etc. The first assumption does not hold in our case, while robustness to distortions is only necessary to a very limited degree in our application, as the content to be matched is captured and processed under very similar conditions. Another application of near-duplicate detection is topic tracking of news stories over time as they develop. The problem of matching video segments can be transformed into a problem of matching sequences of feature vectors extracted from the video segments. Two classes of suitable distance measures for sequences of these feature vectors have been proposed.

One is based on the Dynamic Time Warping (DTW) paradigm [5], which tries to align the samples of the sequences so that the temporal order is kept but the distance is globally minimized. The approach has been applied to the detection of repeated takes in rushes video [4]. The authors of [8] propose a method that is conceptually very similar to DTW but includes further strict constraints, e.g. it is assumed that start

In Proceedings of 1st International Workshop on Automated Information Extraction in Media Production at DEXA 2008, Turin, IT.

and end of the two video segments are temporally aligned. The distance measure Nearest Feature Line (NFL) [12] does not align samples of the two sequences but calculates the nearest point as the intersection of a line that is orthogonal to the line in feature space between two samples and passes through a sample of the other sequence.

The other class of distance measures is based on the concept of the edit distance between strings, i.e. the cost of inserting, deleting or replacing samples in the sequence. The authors of [1] propose such a measure called *vString* edit distance. The values of the feature vectors are mapped to a set of discrete symbols and three new edit operations are introduced: fusion/fission, swapping and insertion/deletion of shot boundaries. The drawbacks are that the sequence of feature vectors needs to be mapped to a discrete set of symbols and that operations such as fission/fusion need to be modeled separately. The Longest Common Subsequence (LCSS) model is a variant of the edit distance, supporting gaps in the match. It has been applied to measuring the distance between trajectories in 2D space [9] and it has been shown that it perform better than other methods for this problem [11]. In [2] a LCSS based measure has been applied to detecting and clustering takes of the same scene in rushes material.

3. Three distance measures

In this paper we compare three measures for the distance of feature sequences of video segments that are capable of discriminating different action taking place even if the scene or the objects are visually similar, but tolerate small variations in object placement and timing, including insertions and omissions. One is based on Dynamic Time Warping, the other two are variants of Longest Common Subsequence.

In the following, $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_n)$ denote the sequences of features to be matched. The length and sampling rates of the two sequences may be different. a_i and b_i are vectors created from chaining K feature vectors. $a_{i,k}$ denotes the part of vector a_i representing feature k. $\mathrm{d}_k(a, b)$ denotes a specific distance function for feature k.

3.1 Dynamic Time Warping

Dynamic Time Warping (DTW) approaches [5] try to align two sequences so that the temporal order is kept but the distance is globally minimized. Each sample of one sequence must be aligned to a sample of the other sequence. The standard implementation builds a distance matrix of size $m \times n$. Then an optimal path through the distance matrix (i.e. from $(1, 1)$ to (m, n) is found by moving either one step in horizontal, vertical or diagonal direction in order to minimally increase the distance. The only modification in our implementation is that the distance between two elements a_i and b_j, which is the Euclidean distance in the standard implementation, is replaced by

$$\mathrm{d}(a_i, b_j) = \sum_{k=1}^{K} w_k \mathrm{d}_k(a_{i,k}, b_{i,k})$$

where w_k is the relative weight of feature k.

Among the proposed improvements of the DTW algorithm are FastDTW [7] and derivative DTW [3]. Both are based on the idea of only matching elements that have a certain offset in the position in the sequence. This assumption does not generally hold in our application, thus these approaches are not used.

3.2 Longest Common Subsequence with threshold parameter

The authors of [9] apply the LCSS algorithm to matching trajectories in 2D space and introduce the following thresholds: a real number ϵ that defines the matching threshold between the non-discrete elements of the sequences and an integer δ that defines the maximum offset in the positions to be matched. In order to adapt the LCSS algorithm to matching takes the following modifications are made. As each element of the sequence is a multidimensional feature vector, a vector $\theta_{sim} = (\epsilon_1, \ldots, \epsilon_K$ is defined, that contains the matching thresholds for all features. Similarly a vector $W = (w_1, \ldots, w_K)$ representing the relative weights ($\sum_k w_k = 1$) of the features is introduced. The offset δ introduced in [9] is not relevant for this problem, as the matching subsequences can be anywhere in the parts. Instead the maximum gap size γ of the subsequence is introduced as constraint. Note that the longest common subsequence does not necessarily fulfill the condition of having only gaps $\leq \gamma$, so instead of just finding the LCSS all sufficiently long subsequences (length $> \theta_{len}$) with gaps no longer than γ need to be found. The method is described in detail in [2].

3.3 Longest Common Subsequence with sigmoid matching function

In [10] an extension of the LCSS method described in [9] is proposed, which eliminates the threshold parameter ϵ. In the original algorithm, the length of the match is increased by one, if the distance is below the threshold. In the extension, the length of a match is increased by the value of a sigmoid weighting function $\sigma(x) = 1/(1 + e^{-px})$, where p is a scaling parameter and x is calculated from the feature distance normalized by the minimum of the standard deviations of the two sequences to be matched.

When applying this extension to matching feature sequences, one has to take into account that each part of the

feature vector $a_{i,k}$ may need to be matched with a different distance function, which may not even be metric. In addition, determining the standard deviation of a feature sequence may not be trivial. In order not to restrict the possible set of features, for each feature the medoid of a sequence is determined as:

$$\text{medoid}(A,k) = \underset{i}{\text{argmin}} \sum_{j=1}^{m} \text{d}_k(a_{i,k}, a_{j,k}),\ j \neq i$$

The standard deviation can then be defined as

$$\text{std}(A,k) = \sqrt{\frac{1}{m}\sum_{i=1}^{m} \text{d}_k(a_{i,k}, \text{medoid}(A,k))^2}$$

and can be plugged into the matching function as described in [10]. As the threshold based on the minimum of the standard deviations has been found to be very restrictive, the maximum of the standard deviations of the two sequences instead is used instead. Note that due to the sigmoid function samples with larger distances are still weighted very low. The other modifications of the LCSS algorithm are the same as described in [2].

4. Evaluation

We evaluate the three distance measures on six randomly selected videos of the TRECVID 2007 BBC rushes summarization data set [6] in order to cluster takes of the same scene, shot from the same camera position. The videos cover different types For the videos manually annotated ground truth has been produced. No discrimination between complete and partial takes has been made. Shots of the video containing only test patters such as color bars or monochrome frames have been excluded from the test data.

The proposed algorithm has been evaluated on a subset of the TRECVID 2007 BBC rushes test data set [6]. The subset consists of six randomly selected videos out of this data set (in total 3 hours, about 14% of the complete set). For this subset ground truth has been manually annotated by identifying the set of scenes and the takes of each scene. All takes of a scene have been shot from the same camera position.

The task is to match the segments from an input segmentation (shots or subshots, as a shot may contain several takes) and to decide which segments (or parts of segments) show takes of the same scene and thus should be clustered. As earlier work has led to the assumption that the performance of the clustering has a strong dependence on the quality of the input segmentation, we evaluate the distance measure with three input segmentations: two outputs of automatic segmentation algorithms (one tending to under-, the other tending to oversegmentation) and a manually created ground truth segmentation. Note that the ground truth segmentation does not contain just the matching segments, but segments corresponding to takes and all material around one take, i.e. including also parts of the take that do not appear in any other takes and sometimes non-scripted material such set up or comments by the director.

The following parameters are used for evaluation. The feature sequences contain feature vectors with MPEG-7 ColorLayout (CL) and EdgeHistogram (EH) descriptors, extracted from every 10^{th} frame of the video and the average visual activity (VA) in of the 10 frames around this position. The relative feature weights are set to $w_{CL} = 0.5$, $w_{EH} = 0.2$ and $w_{VA} = 0.3$. The minimum length θ_{len} is 25 frames, the gap size γ is 20 frames (the videos in the test have a frame rate of 25). These parameters are the same for all three algorithms. For the LCSS with threshold $\theta_{sim} = (0.03, 0.03, 0.03)^T$. For the LCSS with sigmoid function and the parameter p is set to 0.25.

In order to evaluate the quality of the result, the overlap between take clusters in the result in the ground truth has been determined as described in [2]. Precision and recall are calculated based on the number of correctly and incorrectly assigned takes.

4.1 Comparison of distance measures on ground truth segmentation

Table 1 shows the evaluation results for all of the videos using the ground truth segmentation as input. The number of takes resulting from matching is similar to the ground truth for all methods, the LCSS method with sigmoid function tends to yield slightly less takes. Note that the LCSS methods can produce a higher number of takes than there are parts in the input segmentation as they might find several separate matching segments.

The number of scenes resulting from the LCSS methods is slightly higher than in the ground truth, less so for the variant with the sigmoid function. The DTW method yields a very low number of scenes. As the DTW distance measure has to find a match for every sample in the sequences, even partly good matches are spoiled and the measure becomes less discriminative.

The LCSS methods yield always significantly better precision and recall rates than the DTW method, the LCSS variant with the sigmoid function performs slightly better than the one with the fixed threshold, but not on all of the videos. In contrast to the other methods, the LCSS variant with the sigmoid function yields significantly higher precision than recall, but still better recall than the other methods.

Method	MRS07063	MRS025913	MRS044731	MRS144760	MRS157475	MS216210	Mean	Median
Number of takes								
GT	26	28	34	24	36	26	29.00	27.00
DTW	26	27	34	24	36	26	28.83	26.50
LCSS t	28	27	34	24	38	23	29.00	27.50
LCSS s	22	28	26	20	35	19	25.00	24.00
Number of scenes								
GT	6	8	7	6	8	7	7.00	7.00
DTW	2	2	1	3	2	3	2.17	2.00
LCSS t	7	10	5	6	10	10	8.00	8.50
LCSS s	6	10	6	5	10	7	7.33	6.50
Precision								
DTW	0.2692	0.4074	0.2059	0.5833	0.3611	0.5385	0.3942	0.3843
LCSS t	0.8214	0.7407	0.5294	0.7917	0.6316	0.6957	0.7017	0.7182
LCSS s	0.9091	0.7143	0.8846	0.9000	0.7714	0.8947	0.8457	0.8897
Recall								
DTW	0.2692	0.3929	0.2059	0.5833	0.3611	0.5385	0.3918	0.3770
LCSS t	0.8846	0.7143	0.5294	0.7917	0.6667	0.6154	0.7003	0.6905
LCSS s	0.7692	0.7143	0.6765	0.7500	0.7500	0.6538	0.7190	0.7321

Table 1. Overview of results of the three algorithms on each of the videos and mean/median of the results, using the ground truth segmentation as input (GT ... ground truth, LCSS t ... LCSS method with threshold parameter, LCSS s ... LCSS method with sigmoid function).

4.2 Comparison of distance measures on real-world segmentations

The influence of the input segmentation on the result is shown in Table 2. A feature sequence of one input segment corresponds to a feature sequence to be matched. Segmentation method 1 tends to produce undersegmentation, Segmentation method 2 produces oversegmentation. It can be seen that segmentation errors degrade the performance of all methods, but to a different extent. The number of segments produced by the DTW method increases significantly in case of oversegmentation, the number of clusters is further reduced and both precision and recall are decreased. For the LCSS method with threshold the number of takes increases and decreases with the number of segments in the input, but only moderately. The number of clusters is decreased for both real-world segmentations. In the case of oversegmentation, both precision and recall are decreased, in the case of undersegmentation precision is slightly increased, but recall further decreased. The LCSS variant with sigmoid functions shows a similar behavior, but it tends to be more restrictive, thus having higher precision rates for both of the real-world segmentations but strongly decreased recall.

5. Conclusion

We have compared three distance measures for feature sequences of videos, using different segmentations as input. The results show that the methods based on the Longest Common Subsequence (LCSS) model perform better. On ground truth segmentation a variant of LCSS that does not need a threshold parameter but determines weighting from the input sequences performs best. In the case of real-world segmentations this method shows higher precision, but significantly reduced recall rates, while the variant with threshold parameter still has acceptable performance even on erroneous input segmentations.

6. Acknowledgements

The author would like to thank Felix Lee, Hannes Fassold, Harald Stiegler, Herwig Rehatschek, Georg Thallinger and Werner Haas for their feedback and support.

The research leading to this paper was partially supported by the European Commission under the contracts FP6-045032, "Search Environments for Media – SEMEDIA" (`http://www.semedia.org`) and FP6-027026, "Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content – K-Space" (`http://www.k-space.eu`).

	Method	Ground truth seg.		Segmentation 1		Segmentation 2	
		Mean	Median	Mean	Median	Mean	Median
Number of takes	DTW	28.83	26.50	36.00	31.00	75.00	64.00
	LCSS t	29.00	27.50	22.67	23.00	31.00	29.50
	LCSS s	25.00	24.00	11.67	11.00	12.17	12.00
Number of scenes	DTW	2.17	2.00	1.50	1.50	1.67	2.00
	LCSS t	8.00	8.50	6.33	6.00	7.50	7.00
	LCSS s	7.33	6.50	4.33	4.00	4.33	4.00
Precision	DTW	0.3942	0.3843	0.2474	0.2614	0.1332	0.1143
	LCSS t	0.7017	0.7182	0.7093	0.7321	0.6049	0.5763
	LCSS s	0.8457	0.8897	0.9058	0.9762	0.8834	0.9706
Recall	DTW	0.3918	0.3770	0.2660	0.2712	0.3052	0.2418
	LCSS t	0.7003	0.6905	0.5603	0.5000	0.6377	0.6346
	LCSS s	0.7190	0.7321	0.3802	0.3205	0.3332	0.3407

Table 2. Mean and median results over the 6 videos using different input segmentations (LCSS t . . . LCSS method with threshold parameter, LCSS s . . . LCSS method with sigmoid function).

References

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

[2] W. Bailer, F. Lee, and G. Thallinger. Detecting and clustering multiple takes of one scene. In *MMM*, pages 80–89, Kyoto, JP, Jan. 2008.

[3] E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *First SIAM International Conference on Data Mining (SDM'2001)*, 2001.

[4] J. Kleban, A. Sarkar, E. Moxley, S. Mangiat, S. Joshi, T. Kuo, and B. S. Manjunath. Feature fusion and redundancy pruning for rush video summarization. In *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pages 84–88, New York, NY, USA, 2007. ACM.

[5] C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal*, 60(7):1389–1409, Sept. 1981.

[6] P. Over, A. F. Smeaton, and P. Kelly. The TRECVID 2007 BBC rushes summarization evaluation pilot. In *Proceedings of the TRECVID Workshop on Video Summarization (TVS'07)*, pages 1–15, New York, NY, September 2007. ACM Press.

[7] S. Salvador and P. Chan. FastDTW: Toward accurate dynamic time warping in linear time and space. In *Proceedings of 3rd Workshop on Mining Temporal and Sequential Data at ACM KDD'04*, Aug. 2004.

[8] Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge. A framework for measuring video similarity and its application tovideo query by example. In *Proceedings of International Conference on Image Processing*, volume 2, pages 106–110, Kobe, JP, Oct. 1999.

[9] M. Vlachos, G. Kollios, and D. Gunopoulos. Discovering similar multidimensional trajectories. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*, pages 673–684, San Jose, CA, USA, 2002. IEEE Computer Society.

[10] M. Vlachos, G. Kollios, and D. Gunopulos. Elastic translation invariant matching of trajectories. *Mach. Learn.*, 58(2-3):301–334, 2005.

[11] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 1135–1138, Washington, DC, USA, 2006. IEEE Computer Society.

[12] L. Zhao, W. Qi, S. Z. Li, S.-Q. Yang, and H. J. Zhang. Keyframe extraction and shot retrieval using nearest feature line (NFL). In *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, pages 217–220, New York, NY, USA, 2000. ACM.