Audiovisual production invariant searching

Siba Haidar[*] — Philippe Joly[*] — Bilal Chebaro[**]

[*] Institut de Recherche en Informatique de Toulouse
118, route de Narbonne - 31062 Toulouse, France
{shaidar, joly}@irit.fr

[**] Université Libanaise, Faculté de Sciences section 1
bchebaro@ul.edu.lb

RÉSUMÉ. *La recherche de l'information non textuelle est un point fondamental dans l'industrie audiovisuelle où les besoins d'outils pour manipuler des contenus multimédia sont importants et diversifiés. Dans les documents vidéo, l'extraction de signature de style est un procédé extrêmement intéressant, puisqu'il fournit une nouvelle caractéristique pour la classification de contenus. Les documents vidéo peuvent avoir des caractéristiques et des propriétés très différentes. Cependant, on peut identifier des points communs à toutes les émissions politiques, ou toutes les retransmissions de matchs de football, ou encore tous les films réalisés par un même réalisateur. Ces points communs sont ce que nous appelons "invariants de production". Un invariant de production caractérise un document ou une série de documents appartenant à une même "collection", ou tourné par un même réalisateur, ou produit suivant les mêmes directives.*

Dans ce papier, nous proposons une transcription formelle de ce que nous appelons "un invariant de production" dans un segment vidéo à travers l'étude de l'évolution des caractéristiques de bas niveau. Nous proposons un algorithme pour l'extraction de segments invariants, applicable sur tout document audiovisuel, indépendamment de la nature des caractéristiques, de leur sens, et du type ou de la durée de l'invariant.

ABSTRACT. *Information searching in non-textual media is a fundamental point of interest, especially in the audiovisual industry where there is still an important need of tools for manipulating multimedia contents. In video documents, the style signature extraction is a highly interesting process since it provides a new feature for contents classification. Video documents may have very different characteristics and properties. However, we all agree that there are some common points between all political programs, or all football matches, or, time to time, between all the movies realized by a given director. These common points are what we call "invariants". An "invariant of production" characterizes a document or a set of documents belonging to a same "collection", of a same director, or produced following the same set of guidelines.*

In this paper, we present a hypothetical definition of what we call production invariant in a video segment. We propose an algorithm for the invariant segment extraction, applicable on all video features independently with the feature nature and meaning, and with this invariant length or type.

MOTS-CLÉS : *indexation audiovisuelle, comparaison de caractéristiques, extraction de caractéristiques vidéo.*

KEYWORDS: *audiovisual indexing, feature comparison, video feature extraction.*

1. Introduction

Video feature extraction is used for segmenting, classifying and indexing video documents. Video content analysis is a first step before applying information retrieval and browsing tools in order to satisfy some queries on audiovisual content.

Many research works were made to extract meaningful information on production styles, in order to classify them and to construct robust summaries (Yahiaoui, 2003). New description strategies were introduced allowing creating ToCs (Tables of Content) and index of a video sequence. (Pinsach, 2003) focuses on the semantic characterization of videos sequences in order to enrich the ToC and Index structures.

The use of extracted video feature to find what could be *production invariant* relative to a given set of video document is very important to classifying semantic scenes and creating ToCs and Index. An invariant of production, as we define it, is what characterizes a document or a set of documents belonging to a same "collection", of a same director, or produced following the same set of guidelines. Once defined and extracted for a set of audiovisual documents, these invariants are useful in various operations concerning documents' semantic analysis. Since invariants characterize video segments, they can be used later to identify or localize them. Another benefit we can see in "invariants", is the validation of production charts, i.e., a director can use them to follow up the evolution of his realizations and see whether or not they do apply his rules and guidelines.

We consider that these production invariants will appear as the repetition of the same sequence of values of different low-level features extracted from two different streams or two different parts of a same stream. The size of these schemas may vary, and all the extracted features are not necessarily involved in the expression of these sequences. It can be for example the iteration of some specific values when the anchor frame appears in a TV News program, or at the beginning of a same game show broadcasted on two different days.

To localize and extract a production invariant, we rely on the set of all possible features characterizing an audiovisual document (Adami *and al.*, 2002) which can be extracted and quantized to build numeric time series. Then we consider the invariant search as a segment seen as a common- or a similar- or even as a repetitive sequence of values, from the point of view of one or multiple features.

In this context, we need to determine whether two given features (or more exactly their time series) display a similar behavior. The problem is interesting (and difficult) because, unlike traditional tools for audiovisual content querying, we do not have, a priori, a sample or model of what we are looking for. This means we do not know previously what we are searching for, where it could be or even what dimension or length it could have.

In this paper, we present a simple algorithm for invariant video content identification, based on a two-level approach. It requires neither any previous knowledge about the feature’s nature or its behavior, nor user intervention in order to define or change thresholds or filters, to produce a result. The algorithm automatically accommodates thresholds and filters to feature general properties. This permits it to be applied on different types of time series.

Since we propose a new algorithm for comparing two time series, we quickly describe the state-of-art of similarity measures and indexing techniques that have been proposed for time series analysis.

The main idea behind similarity measures is that they should allow imprecise matches. There are several applications of such measures. For example, they can be used to cluster the different time series into similar groups, or to classify a time series based on a set of known examples. Another point of interest is the indexing problem which can be formalized as : "given a set of time series Q, prepare an index offline such that given a query series q, the time series in Q that are most similar to q can be reported quickly" (Gunopulos *and al.*, 2000). All similarity measures intend to bypass obstacles, such as, the subsequence similarity problem, the rule discovery problem, and clustering problems which have to be overcome in our approach. Furthermore, we consider that, as well as accuracy, efficiency (in terms of computational cost) is a challenging issue.

Euclidean Similarity Measure views each sequence as a point in n-dimensional Euclidean space (n=length of sequence) and defines the similarity measure between X and Y as Lp(X,Y). Although it is very easy to compute, it does allow neither noise nor short-term fluctuations, nor temporal shifts. Measures based on transformation rules, like for example moving average is a well known technique for smoothening time sequences. Combined with Euclidean distance, it intuitively produces similarity results but does not fit our requirements. For example, it does not preserve relevant and meaningful peaks, like the ones we can observe in the evolution of action/motion features.

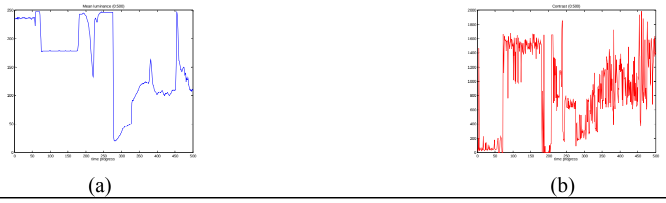
Dynamic time-warping based matching is another popular technique in the context of speech processing (Sakoe *and al.*, 1978), sequence comparison (Erickson *and al.*, 2002), and shape matching (McConnell, 1991). This method has been used in (Guttman, 1984) to match a given pattern in time-series data. The essential idea is to match one dimensional pattern while allowing for local stretching of the time parameterization (Brendt *and al.*, 1994). It is a robust measure that allows non-matching gaps, amplitude scaling, and offset translation.

Longest Common Subsequence Measures: based on the edit distance (Bozkaya *and al.*, 1997), (Ganascia, 2002) works on sequences with slightly different length. It considers two sequences to be similar if they have enough non-overlapping time-ordered pairs of subsequences that are similar. The algorithm consists of finding all atomic similar subsequence pairs, which is achieved by a spatial self-join over the sets of all atomic windows. Edit distance is also used for approximate text matching, based on dynamic programming. Some methods are inspired from algorithms for fast text searching (Argawal *and al.*, 1995), (Ganascia, 2002), like finding text subsequence that approximately matches a given string. Text sequences normally consist of a few discrete symbols as opposed to continuous numbers that makes the similarity measures and the search methods quite different.

2. Multi-level analysis coupling using morpho-math filters

All the similarity measures proposed in the domain of time series do not intend to merge the result of multiple comparisons. They are supposed to be applied on one type of time series at a time. Thus, parameterization issue is not a problem for them; it can be fixed once on the beginning of the processing depending on the type of studied time series. In our case, a production invariant is identified after combining all studied features results.

As we will show, features extracted from an audiovisual document do not have a comparable evolution.



(a) (b)

Figure 1. *The mean luminance feature smoothness (a) relative to the eventful color contrast feature (b). Both are extracted from the same audiovisual segment.*

For example, we have defined a measure of the color contrast which is obtained here by iteratively measuring the distance between the two main dominant colors, given by the following formula:

$$contrast = \left|L_1 - L_2\right| + \mathrm{distcirc}\left(H_1, H_2\right) \times \log\left(\frac{S_1 + S_2}{2}\right) \qquad \textbf{[1]}$$

Where L_1, L_2 are respectively the luminance of the two dominant colors, H_1,H_2 the hue, and S_1, S_2 the saturation, and where district(x,y) is the circular distance between x and y, knowing that x and y vary inside certain boundaries

This feature is very noisy and eventful (Fig.1.a), and presents many peaks. On the contrary, a feature with a very different behavior is the mean luminance, which calculates the average of pixels' luminance in each frame, is very smooth and calm.

As it appears in fig.1, no normalization was applied on features, because until now, all treatments applied are relative to each feature and so there is no meaning of a scale comparison.

Considering this fact, no feature nature discrimination is allowed; the comparison algorithm is applied uniformly on all the time series, despite their smoothness. However, we accommodate the processing by varying the thresholds relatively to each sequence property.

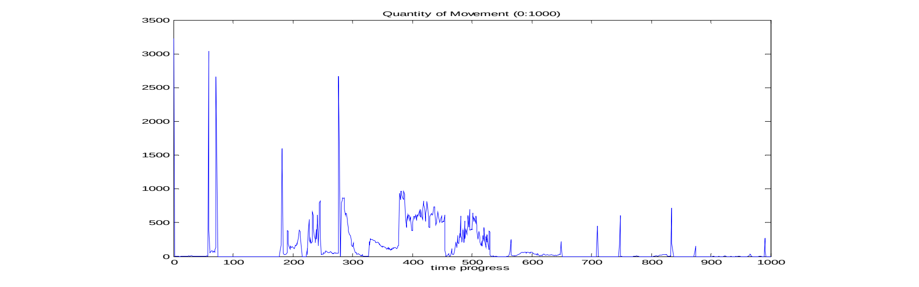


Figure 2. *Meaningful peaks in the quantity of movement feature; they announce scene change. They are either sudden peaks designating a cut transition, or progressive when occurs a cross-dissolve.*

Also, we allow no smoothing considering that peaks carry important information related with an event happening during the video, like a lightning effect or a sudden movement or even a view or a scene change (see fig.2 for example). At the matter of fact, only a few peaks may be due to noise.

For this reason, we proposed a new algorithm, able to set automatically its parameters, to conform features special properties.

2.1. *Matching process*

Our sequence matching method, like most of the matching methods, in an environment with a large set of data sequences, works in two phases. In the first phase, only a finite number of data sequences are kept after a filtering process. We consider that these sequences are matching candidates. In the second phase, all candidate sequences are verified for the actual matching using a morphomath comparison filter.

Before generalizing and giving a definition for matching sequences in scientific experiments, we would like to address the following points for motivation:

- The relative times that the corresponding samples are taken are almost the same in both sequences. This means that lengths of sequences to be matched should be close to each other.

- Two sequences can be considered as matching (or similar) if the majority of their segments elements match.

Definition 2.1 *Matching sequences*: Two subsequences of length l are considered to be similar if their similarity covering measure bypasses a certain threshold.

Definition 2.2 *Covering:* The covering is the percentage of sub sequences touched by the morphomath coupling process. It is given by the following formula;

$$cover(I_i, J_j) = \frac{nb(coupled\_subsequences)}{total\_nb\_subsequences} \qquad \textbf{[2]}$$

Where nb(coupled_subsequences) is the number of subsequences coupled a least once in the process of similarity comparison

The choice of the similarity threshold covering depends on the accuracy requested for the similarity measure.

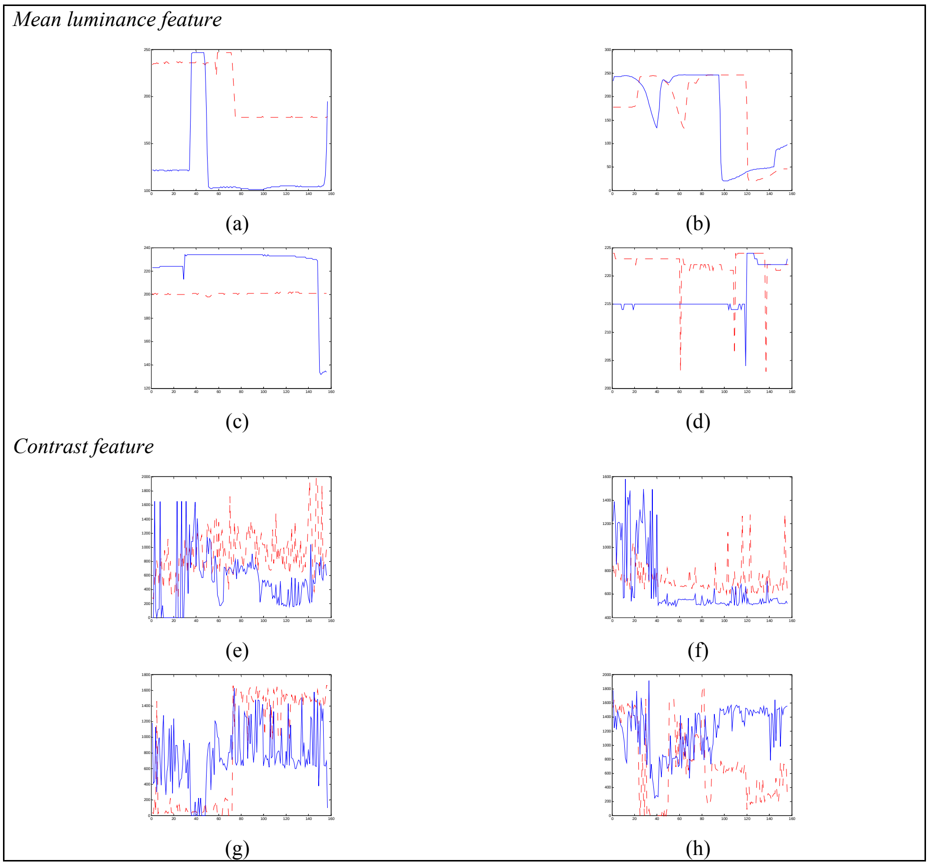
The algorithm proceeds in a dichotomous approach in order to avoid useless comparisons. No continuation in depth is made unless suspected resemblance is possible. We introduce here an algorithm that we call the Recursive Quadratic Intersection (Algo.1).

```
RQI(a,b,t)
{
        % a & b are of similar length
        % t is the desired length of candidates
        if [min(a),max(a)]∩[min(b),max(b)]≠∅
                if length(a)>=t & length(b)>=t
                        divide equally a into $a_1$ and $a_2$
                        divide equally b into $b_1$ and $b_2$
                        RQI($a_1$, $b_1$, t)
                        RQI($a_2$, $b_1$, t)
                        RQI($a_1$, $b_2$, t)
                        RQI($a_2$, $b_2$, t)
                else add (a,b) to potentialCandidates
        return potentialCandidates
}
```

Algo.1 *The extraction of potential candidates for resemblance from two sequences of similar length.*

The result is a set of candidate intervals all of which have the requested length l so that: $\frac{t}{2} \le l < t$.

$$potentialCandidates = R = \{(I_i, J_j), I_i \subset a \wedge J_j \subset b \wedge I_i \approx J_j \wedge \text{length}(I_i) = \text{length}(J_j) = l\}$$



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3. *Extraction of hypothesized candidates found by applying the algorithm using two different features, for t=20 s environ***.**

Once potential candidates are filtered, given two sequences, we shall proceed to the resemblance verification by applying the morphological multi-scale comparison using a structuring element of size α to each couple *(I_i,J_j)*. The value of α is determined on the base of the feature variation. (Algo. 2)

```
verifySimilarity($I_i$, $J_j$, $\alpha$){
{
        % $I_i$ & $J_j$ are of similar length
        % $\alpha$ is the morphomath filter
        if covering($I_i$, $J_j$, $\alpha$) > P % similarity threshold
                  add ($I_i$, $J_j$) to similarCandidates
        return similarCandidates
}
coveringCalculation($I_i$, $J_j$, $\alpha$)
{
        if length($I_i$)&length($J_j$)< $\alpha$ & $[e(I_i), d(I_i)] \cap [e(J_j), d(J_j)] \neq \emptyset$
                  CountOccurrence if first time coupled
        else
                  divide $I_i$ into $I_{i1}$ & $I_{i2}$
                  divide $J_j$ into $J_{j1}$ & $J_{j2}$
                  for each couple:
                  if $[e(I_{ii}), d(I_{ii})] \cap [e(J_{jj}), d(J_{jj})] \neq \emptyset$
                      coveringCalculation($I_{ii}$, $J_{jj}$, $\alpha$)
        At the end compute as shown in equation [2]
}
```

Algo.2 *Similarity verification using morphomath filter (structuring element) which length is adapted to the sequences coefficient of variation.*

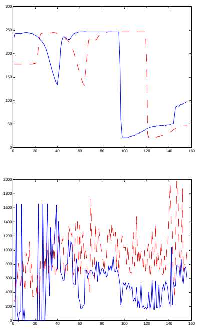
Here we used *e(x)* and *d(x)* to designate, respectively, the morphomathematical erosion and dilation of a sequence *x*, computed in a one-dimensional space, with the filter size α. In our experiments, we used the minimum and maximum, as classical erosion and dilation.

The reason we have chosen the morphomathematical operators in subsequences comparison is that these operators perform fast and are well adapted to handle value fluctuation and imprecision inside a segment of filter size at the same time.

The filter α is chosen relatively for each sequence (feature), based on the following reasoning. The more the sequence presents variations in its evolution, the bigger will be α in order to allow matching of two sequences. In the contrary, the smoother is the shape of curves; the more precise we have to be in comparing final sequences, thus the smaller the morphomath filter will be.

The filter depends of the variations of the time series, which is defined as follows. If S_x is the standard deviation of a set of samples x_i and m its mean, then $V=S_x/m$, (Weisstein, 1999). So, $\alpha=f(V)$.

This means the morphomath filter is related to the coefficient of variation. This work hypothesis is to be validated by further works and experiments. For the time being, the relation between those two observed parameters is still empirical. The threshold varies between two boundaries, which assert our hypothesis.



$\alpha=2$	the covering is 83%
$\alpha=4$	the covering is 93%
$\alpha=6$ and up	the covering is 100%

CoefficientOfVariation=0.1865

$\alpha=2$	the covering is 90%
$\alpha=4$	the covering is 91%
$\alpha=6$	the covering is 95%
$\alpha=12$	the covering is 97%
$\alpha=20$ and up	the covering is 100%

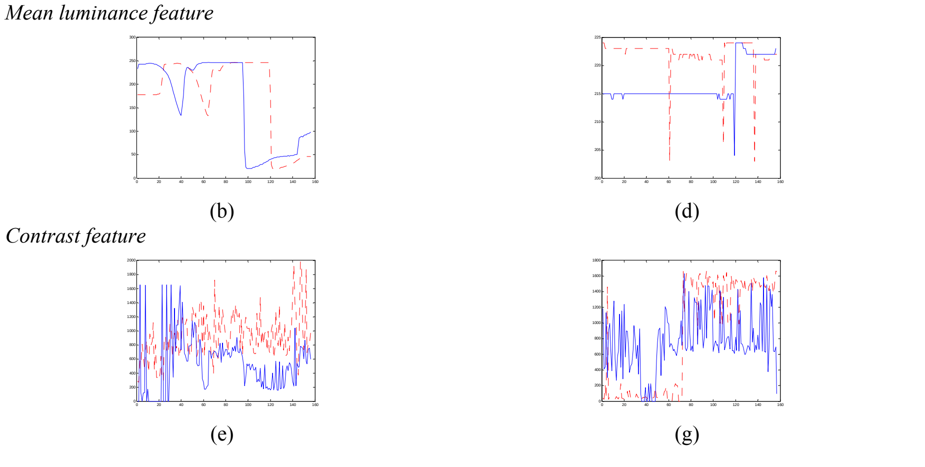
CoefficientOfVariation=0.5095

Figure 4. *The more the coefficient of variation is bigger, the more the filter has to be wide in boundary limits.*

The final result is a set S where

$$S = similarCandidates \subset R = potentialCandidates\,.$$

Note that, the S set contains segments of length l. $\frac{t}{2} \le l < t$.



Mean luminance feature

(b)

(d)

Contrast feature

(e)

(g)

Figure 5. *Referring to fig.3, only (b) and (d) and (e) and (g) where kept (with many others) after similarity verification; Here similarity cover threshold was set to 95% which is not a strict value***.**

Considering this, for a given feature, couples in S, of length l, are similar. But one should not forget that, the goal of our similarity search is to find the similar subsequences despite of their length.

In consequence, potential invariant could have a length of 1 second up to 30 seconds or even more in a video content. That’s why avoiding loosing any precious information, we preserve all possible similar subsequence for later filtering. Thus, parameter t must not be fixed and must vary between min and max boundaries.

```
varyingLength(a,b,tMax,tMin)
{
        % a & b are of similar length
        while(length(a)&length(b)> tMax)
                divide a into $a_1$ & $a_2$
                divide b into $b_1$ & $b_2$
        for each couple:(x,y) compare(x,y)
}

compare(x,y)
{
        if potentiallySimilar(x,y)
                verifySimilarity(x,y, $\alpha$)
        else if length>tMin
                divide by two: $x_1$, $x_2$, $y_1$, $y_2$
                compare($x_i$, $y_j$, $\alpha$), …
}
```

Algo.3 *Varying the length of searched couples.*

In our experiments, we have fixed for t a minimal value of one second, i.e. 24 images, and maximal one of 30 seconds, considering that an advertisement can last this long, and may be considered as invariant.

The above algorithm works on deep down comparison. It continues to compare until it finds all possible similar segments of different lengths. This time, we will obtain variable length couples judged similar by the feature in question.

In the next paragraph, we demonstrate how we filter significant couples based on the combination of multiple features.

2.2. *Invariant filtering based on merging criteria*

The result of the above algorithm, when applied to one feature, i.e. one time series, can be viewed as the union of several sets each of which has the form of S for a certain scale t (Eq. 3).

$$FS = \bigcup_t \left\{ \left(I_i, J_j\right), I_i \subset a \wedge J_j \subset b \wedge I_i \cong J_j \wedge \text{length}\left(I_i\right) = \text{length}\left(J_j\right) = l \wedge \frac{t}{2} \leq l < t \right\} \qquad \textbf{[3]}$$

Where t varies between the boundaries tMax and tMin of what is considered as a reasonable invariant size. For example [1s, 30s].

When applying this general algorithm to all the extracted audiovisual features, we obtain as many sets as the number of features. Certain, the result need to be filtered. Only pertinent information will be kept. Therefore, it is necessary to consider the following cases:

– First, in general, one tiny common segment is meaningless unless a large number of features say the contrary

– in a second hand, a large segment *(I_i, J_j),* said to be invariant, even with only a few number of features (it can be one feature) is most probably an production invariant.

In conclusion, invariants will be filtered relatively to their size and the number of features agreed on their common or invariant property.

The first intuitive method to combine criteria is to extract all common couples of equal size from all features. Even here; we obtain results; like the pre-programmed video sequences that are always inserted in certain programs. Note that, the same segment, diffused several times, will be highly influenced by the noise and timing variations which make it not identical any more.

Then, we can proceed to all possible combining methods. We can start by varying the number of features combined, with or without discrimination, up to varying the compared size of segments, i.e. comparison by intersection (or inclusion) of segments and not only by equality. Here, we can vary the reminder rate and precision rate, regarding to the expected results.

The merging criteria remain a point to explore; some of the results obtained will be shown in the next section, to give an idea of its high interest.

3. Example

One concrete example for the invariant we were able to extract after combining the results for each feature, is the generic extraction without previous model in hand or learning phase. All the features agreed on the invariant property of the generic of a TV game show because it is automatically sent on the beginning of each transmission. It means that we succeeded to catch this invariant despite the noise effects on transmission. In fig. 6, we can see some of the curves (features extracted) relatives to the generic of TV game program (between the black bars), for two consecutives days. The generic evolution is quickly shown in fig. 7 (a, b, c and d).

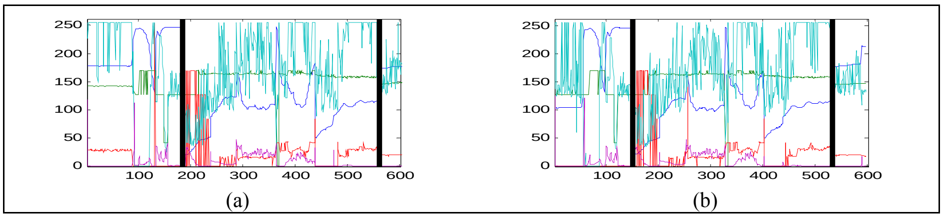


Figure 6. *The show's beginning generic features on on two consecutive days.*

Also, when we change merging criteria, we find that a large number of features detect the monotone aspect of the camera movement as it moves between the animator and participants in a predefined cycle. Although participants change from day to day, invariants were detected regarding to the way the camera moves in some specific shots occurring at specific moments of the game. (fig.7e- h)

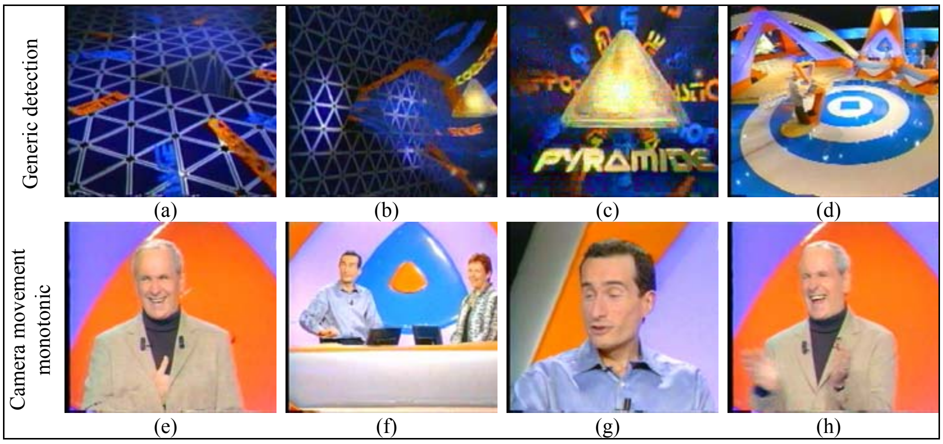


Figure 7. *Two different combining methods (a,b,c,d) and (e,f,g,h) gives two different kind of invariants.*

4. Conclusions and future works

We proposed a method for production invariant extraction from video sequences. The method is based on a hierarchical comparison approach using morphomathematical filters. Given two audiovisual documents, we used fast search techniques able to extract all similar subsequences, of different lengths, and this for each feature characterizing the document.

We have introduced the idea of combining multiple subsequence comparison, of features extracted from audiovisual documents, in order to filter significant results. We can develop some strategies in order to robust invariant detection, based on the length of repeated patterns of values and using other features. This will improve the potential detection of production invariants.

These invariants will be extracted by the analysis of different programs of a same collection. Once they are determined after this learning step, they will be easily detected in any other document of the same kind, or eventually be used to automatically detect an occurrence of an item of this collection. Since we do not have any learning sequence in this first part of the work, we do not use any models to proceed to that kind of detection. But, it already allows us to analyze and to detect repetitions of some specific parts of any given recurrent programs.

We consider that extracted production invariants from real dimension video documents (a whole journey of television broadcasting) will form a set of descriptors we call "Middle Metadata" (MMD). We define the latter as the Metadata layer situated between Low Level features that are concrete and numerically meaning relied to information extracted from audiovisual documents, and Metadata seen as semantic pieces of description adapted to a specific application or to predefined end-user profiles.

In future works, we will study the interest of the normalization of time series before processing, in order to transform feature values, and so to be in a situation to compare these values at any moment along the time series. This will lead us to further study the dependence between the morphomath filter and the variation of time series. While studying this filter, we will evaluate the interest of keeping information about high peaks as shown in fig. 2, and filtering low frequency variations. Later, in order to improve the robustness of invariant extraction, we will combine criteria on the base of audiovisual rules.

Acknowledgement

This work has been conducted on the behalf of the KLIMT-ITEA project and in the PIDOT-CNRS framework.

5. Bibliography

N. Adami, M. Corvaglia and R. Leonardi, "Comparing the quality of multiple descriptions of multimedia documents", *MMSP 2002 Workshop*, St. Thomas (US Virgin Islands), Dec. 2002.

R. Argawal, K. Lin, H. Sawhney and K Shim, ″Fast Similarity Search in the Presence of Noise, Scaling and Translation in Time-Series Databases", Proceedings of the 21st VLDB Conference, Zürich, Switzerland, 1995.

D. J. Berndt and J. Clifford. "Using dynamic time warping to find patterns in time series″, *KDD-94: AAAI Workshop on Knowledge Discovery in Databases,* Seattle, Washington, July, 1994.

T. Bozkaya, N. Yazdani and M. Özsoyoglu, "Matching and Indexing Sequences of Different Length*s*″, *Conference on Information and Knowledge Management, Proceedings of the sixth international conference on Information and knowledge management*, Las Vegas, USA, 1997.

G. Das, D. Gunopulos, and H. Mannila. "Time-series similarity problems and well-separated geometric sets″, *In 13th Annual ACM Symposium on Computational Geometry*. Association for Computing Machinery, 1997.

B. W. Erickson and P. H. Sellers. "Recognition of patterns in genetic sequences". *In D. Sankoff and J. B. Kruskal, editors*, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison. Wesley, MA, 1983.

J-G. Ganascia, ″Extraction of syntatical patterns from parsing trees″, *JADT 2002 : 6es Journées internationales d'Analyse statistique des Données Textuelles. 2002.*

D. Guégan, ″Séries chronologiques non linéaires à temps discret″, Book, Statistiques Mathématiques et Probabilité, Economica, 1994.

D. Gunopulos and G. Das, ″Time Series Similarity Measures″*, Sixth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Boston, MA, USA, 2000.

A. Guttman, ″R-trees: a dynamic index structure for spatial searching", ACM SIGMOD International Conference on Management of Data, Boston, June, 1984.

E. Keogh and P. Smyth, "A probabilistic approach to fast pattern matching in time series databases", Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining. AAAI Press. 1997.

R. McConnell, ″Correlation and dynamic time warping: Two methods for tracking ice floes in SAR images″, *IEEE transactions on Geosciences and Remote Sensing*, 1991.

J. Pinsach, ″Analysis of Video Sequences for Content Description. Table of Content & Index Creation and Scene Classification″, PhD Thesis, University Politècnica de Cataluna, Barceluna, May, 2003.

H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognitio*n″*, *IEEE transactions on Acoustics, Speech and Signal Processing*, 1978.

J. Serra, ″Outils de morphologie mathématique pour le traitement d'images″, Centre de Morphologie Mathématiques, Ecole des Mines de Paris, Fontainebleau, 2000.

E. W. Weisstein, ″CRC Concise Encyclopedia of Mathematics, Second Edition″, CRC Press LLC, Wolfram Research, Inc., 1999-2003.

I. Yahiaoui, ″Construction Automatique de Résumes Vidéos″, PhD Thesis, Télécom Paris, October, 2003.