CHAPTER – 5

VIDEO INDEXING

5.1. INTRODUCTION

Digital video is becoming the rising tide of multimedia. In this chapter we propose a new method to index a video sequence using two methods, the content-based video indexing and meta-data based video indexing.

For effective and interactive exploration of large digital video archives there is a need to index the videos using their visual, audio and textual data. Multimedia technology has been applied to many types of applications and the great amount of multimedia data need to be indexed. The usage of digital video data is very popular today. In particular, video browsing is a necessary activity in many kinds of knowledge.

The amount of video data is growing dramatically. Thus indexing and cataloguing of digital videos are more and more important for retrieval. The best way for indexing video data is content based. In the past, video data are usually described and annotated manually. However this traditional solution is not suitable for the enormous amount of video data. We must find a mechanism that can provide an efficient and flexible solution to illustrate video content. In order to analyse video content we must to segment its content in units. It is possible to do this at two levels:

1. Structural level, and then we divide videos into frames, shots, clips, episodes or scenes.

2. Content level, according to cinematographic properties, motion of the camera, audio properties, motion of a character/object, scenes and stories within a video, etc.

Indexing on video content is possible from two points of view: temporal segmentation and content analysis. The first is the identification of meaningful video segments (as shots, scenes, and episodes); the second is the identification of attributes characterizing regions, objects, and
motions in a video segment. We define segmentation, as the process of breaking down a video into its constituent basic elements that is the shots, and their higher-level aggregates, such as episodes or scenes. There are traditional approaches to performing segmentation composed by the following steps: previewing the whole video, identifying the shots, episodes and scenes and then providing them with their boundaries of textual labels. Since this solution is very time consuming there is a less expensive way, that is to use the edit decision list created by video producers during post-production, but there are few producers that use this method. The detection of shot boundaries is possible either on the raw video stream or on compressed data.

There are two main methods to do this, 1) Cuts detection, where the cut is defined as a clean transition between a shot and the following; it generally corresponds to a curt change in the brightness pattern of two consecutive images 2) Gradual transitions detection, where the change from one shot to another is detected through a number of frames which present some optical effect as fade-in and fade-out, wipes and mattes, etc.

Since a typical segmentation into shots of some types of video (like movies, news and documentaries) produces too many shots (e.g. 600-1500 in a movie) there is the need to build shot aggregates, useful not only for the evaluation of video content, but also for video access at semantic level, for example a sequence of short shots stresses fast action while a sequence of shots with motion, alternated with static shots, stresses dynamics. The shot can be an effective method to segment some formats of video, where it is a useful basis to create new episodes (e.g. in news video), but it is very laborious for video formats where the complete fruition process prevails (as in shot aggregates or episodes).

The development of techniques to support content-based access to archives of digital video information has recently started to receive much attention from the research community. The reason for highlighting the message from the TREC video track that there are now a variety of approaches available for searching and browsing through digital video archives, that these approaches do work, are scalable to larger archives
and can yield useful retrieval performance for users. This has important implications in making digital libraries of video information attainable.

Video indexing consists in describing the content of audiovisual sequences from a video database to allow their retrieval. This concerns television archives, digital libraries, video servers, and digital broadcasting. Like for text document indexing, the purpose is to allow content-based retrieval instead of using only a bibliographic record. Video content includes characters, objects, dialogues, and specific events occurring in a video. Video content can be described at two complementary levels,

- The semantic description, it usually requires human interpreting though some aspects can be assisted by automatic analysis.

- The visual characterisation of images, objects, and also motion, which allows retrieval by example. It is useful to exploit visual features difficult to describe with words.

The work presented here includes Audiovisual Sequences and Multimedia Exploration System. The work involves several complementary fields, knowledge modeling to organise video annotations, databases, high performance parallel architectures, and semi-automatic image analysis. Since we deal with TV archives (films, reports, news, TV programs), the content variety leads us to use the interpretation capacity of the operator, which is very difficult to model. The work on MPEG compressed video, which is compulsory for realistic applications.

The above techniques, either exploit color, motion or texture information in order to provide content-based query capabilities or use eigen value decomposition to reduce the image dimension. Our approach is oriented to extracting a small amount of information, which is sufficient to provide a meaningful representation of a video sequence. This approach not only provides a more efficient way for video indexing, but also results in reducing the storage requirements and thus permits easy management of multimedia databases. An integrated framework for automatic
extraction of characteristic frames has been proposed. The extraction mechanism was based on time variations of the frame feature vectors, generated using color and motion segmentation. However, since different segments may characterize similar frames, the overall procedure was rather sensitive and heavily dependent on the adopted segmentation algorithm. In this chapter, introducing an optimization method for locating a set of minimally correlated feature vectors enhances the frame selection mechanism. Furthermore, a scene selection mechanism is proposed, based on minimization of a distortion criterion for clustering the scene feature vectors.

The traditional keyword annotation approach to accessing image or video information has the drawback that, apart from the large amount of effort for developing annotations, it is not efficient to characterize the rich content of an image or video using only text. For this reason, the MPEG group has recently begun a new standardization phase (MPEG-7) for a multimedia content description interface. This standard will specify a set of content descriptors that can be used to describe any multimedia information. Several tools and algorithms have been proposed in the recent literature for image and video analysis, segmentation or representation, which can be used for the purposes of content-based image retrieval (CBIR). However, most of these prototypes are mainly restricted to still images and cannot be easily extended to video databases since, due to the strong temporal correlation of video frames, performing queries on each video frame is very inefficient.

5.2 STATE OF ART ON VIDEO INDEXING

Description generation: An important concept for the detection of shot aggregates is the keyframe that is a particular frame from the video stream that represents its content or, more usually, a part of it. Higher-level aggregates in a movie can be detected by analysing the similarity between keyframes or repetition of shot keyframes. An example of use of keyframe is in [Lee00], where in order to create an automatic video content description, video is firstly segmented in scenes, that compose the story unit; keyframes are extracted from them and then key features are
produced. Finally descriptors are generated. We summarize this process in Fig. 5.1.

![Diagram showing the process of video content description](image)

**Fig. 5.1 The description generation**

Once a video stream is segmented into its constituent elements, it is necessary that content indexes are set. We create indexes on objects and motions, either on the meaning conveyed by visual primitives.

Indexes on objects are usually extracted from the keyframe (as mentioned above); the key features (information extracted from the keyframe) are used in comparison with primitives (or features) extracted from the query image. The indexes mentioned above are usually full text keywords, or a structured set of concepts, both obtained with human intervention. But it is also possible the use of algorithms in image analysis for automatic extraction of key features. Different types of video need different types of indexes on video content. But we are interested in manual annotation and in particular in visual iconic annotation.
It combines two distinct representations:

- A semantic representation, which is independent from temporal ordering of object actions

- A temporal representation, which establishes specific relationship among objects through their combination and temporal ordering of their actions.

Icons visually represent categories or situations that are in the video, used as visual primitives or compound descriptors. The annotation is usually based on visual languages. An approach particularly suited to describing object spatio-temporal a relationship in a sequence is the iconic annotation by visual examples. These visual examples represent the content of a video segment that will be parsed into a symbolic sentence, according to a special description language. This approach has been used by some authors for its expressiveness and because through it we can generate very detailed descriptions of dynamic content of a video stream. [Arnd89, Del95] discussed symbolic description of motion trajectories for indexing video content through 2D Strings (to represent object spatial arrangement in individual frames) and set theory (to describe changes due to motion). Del Bimbo et al. presented the language Spatio Temporal Logic (STL) in order to represent in a symbolic way spatio-temporal relationship in shot sequences. The basic idea of STL is the spatial assertion, which captures the spatial arrangement of the objects in a scene. Groups of successive frames with equivalent spatial descriptions constitute the states, which in turn are combined through the Boolean connectives and the temporal-until operator. Finally the expression constructed with STL will be parsed in a visual sentence (this mechanism is particularly used in the querying phase).
5.2.1 Content Based Video Indexing

In the earlier stages of information retrieval, the annotation was done in a complete manual fashion. Today, this process is often supported by automatic annotation, which is also referred to as content-based retrieval. The rise of XML as the information exchange format has also its impact on multimedia information retrieval systems.

The semi-structured nature of XML allows integrating several more-or-less structured forms of multimedia annotations in the system, i.e. the database stores a collection of (integrated) XML documents. However, using the XML document structure directly for searching through the document collection is likely to lead to semantic misinterpretations. For example, an XML document contains the structure company $\rightarrow$ director $\rightarrow$ name. When searching for the name of the company, without any knowledge of the semantics of the implied structure the name of the director can, mistakenly, be found. With the focus on document collections where the content is semantically related, it becomes feasible to use a conceptual schema that describes the content of the document collection at a semantical level of abstraction. This approach, defined in the Webspace Method [Avri99] allows the user to formulate complex conceptual queries that exceed the ‘boundary’ of a document. However, using a conceptual search alone is not satisfactory.
A block diagram of the proposed architecture for video content representation is illustrated in Fig.5.3, consisting of four modules; shot cut detection, video sequence analysis, fuzzy classification and key frame extraction.

![Block diagram for video content representation](image)

**Fig. 5.3 Block diagram for video content representation**

Since a video sequence is a collection of different shots, each of which corresponds to a continuous action of a single camera operation, a shot detection algorithm is applied first. It is based on information directly available in the case of intra coded frames of MPEG video sequences, while for the inter-coded ones; it requires a minimal decoding effort, resulting in significant reduction of the required computations.

Video Sequence Analysis: The next step is segmentation of each shot into semantically meaningful objects and extraction of essential information describing those objects. Color and motion segmentation is applied for this purpose, while color and motion information is kept distinct in order to provide a flexible video content representation where each piece of information can be handled separately. The Recursive
Shortest Spanning Tree (RSST) [Morr86] algorithm is our basis for color segmentation. Despite its relative computational complexity, it is considered as one of the most powerful tools for image segmentation. In order to yield faster execution, a new approach is proposed in [Spee98], which recursively applies the RSST algorithm on images of increasing resolution. It is shown that it the exact segment contours can be obtained at the highest resolution level even without knowledge of the image at that level, making it possible to segment frames in MPEG video streams with minimal decoding of a very small percentage of blocks.

![Image](a) ![Image](b)

**Fig. 5.4 Motion segmentation results (a) without, and (b) with smoothing**

Motion segmentation is performed by applying the recursive RSST algorithm at the MPEG block resolution, while motion vector differences are used instead of color differences. We have chosen to exploit the motion vector information that is directly available in MPEG streams, thus eliminating the need for motion analysis altogether. Although an extremely fast implementation is achieved in this way, a post-processing median filtering step for motion field smoothing is indispensable [Doul98]. It is clear from Fig.5.4 that without motion vector smoothing, wrong segmentation results are produced, even in a uniform and almost stationary background. On the contrary, only the actually moving objects are extracted in the case of smoothed motion vectors.
5.2.2 Metadata Based Video Indexing

In order to represent the video structure and Dublin Core descriptors outlined in Fig. 5.1, a suitable schema must be able to support the following:

- Hierarchical structure definitions: The schema must be able to constrain the structure to a precise hierarchy in which complete video documents sit at the top level. These in turn contain sequences, which contain scenes, which contain shots, which contain frames, which contain objects or actors. Fig. 1 illustrates this hierarchy.

- Each level (or class) within the hierarchy must be constrained to possess only specific attributes. In our description scheme, we assume that each layer possesses the 15 simple, optional and extensible DC elements plus a set of class-specific attributes unique to that layer. These represent the set of MPEG-7 descriptors for that class when they become available.

- Element and attribute inheritance: It should be possible to specify sub-classing with inheritance of attributes and elements from the upper to lower classes. In addition, sub-classes should be able to have their own additional attributes and elements. This allows efficient reuse and customization of document schemas.

- Data Typing: It must be possible to constrain the values of attributes to certain data types. Data types supported should include primitive data types as well as Schemes (e.g. SMPTE), enumerated data types, controlled vocabularies, file types (images) and complex data types (e.g. colour histograms, 3D vectors, graphs, RGB values etc.). It should also be possible to specify multiple alternative schemes or data types for a particular attribute.
• Cardinality within attributes should be representable: It must be possible to specify that an attribute can have zero, one or multiple values. Ideally the minimum and maximum number of attributes should also be specifiable e.g. a scene must contain between 2 and 5 shots.

• Spatio-temporal specifications. The Schema must be able to support the specification of temporal characteristics e.g. begin and end time of segments and their duration. Similarly, it should be able to support spatial representation e.g. regions within an image or motion along a line.

• Spatial, temporal and conceptual relations: Spatial relations such as neighboring objects and temporal relations such as sequential or parallel segments should be supported. Given such a relationship between two classes, it should also be possible to constrain specific attribute values of these classes. For example, the start and end times of scenes contained within a sequence, must lie within the start and end time of that sequence.

• Human-readability: It is desirable rather than mandatory that both the schema and the description output from the schema should be human-readable.

• Availability of supporting technologies such as parsers (capable of validating input descriptions), databases and query languages.

5.3 PROPOSED ALGORITHM

Video indexing based on content and meta-data: Video indexing through the use of standard metadata caused a great interest from different research groups, among these the DCMI Moving Pictures Special Interest Group; on its proposal we will base ours. Firstly we need to define our criteria to video segmentation (which we will derive from the analysis of some criteria seen in previous section). Afterwards we will propose for those levels (in which the video is segmented) the corresponding
metadata, whose elements will be just derived from Dublin Core metadata element set. Our proposal on video segmentation is based on modification of scheme showed in fig 5.1, where two video segmentation levels surface: the first level is the scene; the second one is the aggregation of many scenes logically connected. It differs from the story unit. Definition a story unit concept of sequence since scenes connected in a story unit can be also not contiguous, while in the sequence scenes are contiguous. Since such aggregation occurs only at logic level, story units are logical entities, which are constructed through the use of metadata. The advantages of introduction of such entity are:

- It does not have to be physically stored, but it need to be characterized in the system catalog. Consequently it will provide a greatest amount of information without further waste of storage.

- It is a logical aggregate of scenes and then can be characterized by a specific Keyframe.

- It can be defined through the use of metadata, and this approach can be extended also to key-frames and scenes.

- Indexing and querying processes use search engines based on metadata.

*Fig. 5.5 Example of possible spatial relations between A and B*
Table.5.1: Content and Metadata for the example

**KEYFRAME**

<table>
<thead>
<tr>
<th>DC.Description.text</th>
<th>Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC.Relation.IsPartOf</td>
<td>Scene D</td>
</tr>
<tr>
<td>Objects</td>
<td>A, B, C, D</td>
</tr>
<tr>
<td>AREM.X</td>
<td>A&lt;B, A%C, A&lt;D, B%D, C&lt;B,C&lt;D</td>
</tr>
<tr>
<td>AREM.Y</td>
<td>A=B, C/A, C/B, C=D, D/A, D/B</td>
</tr>
</tbody>
</table>

For entities that we chose the following levels of metadata are defined:

1. The first level is for metadata on the whole video (for it we adopt the classical approach using the whole set of Dublin Core metadata) and for the scene (for it we use a subset of the above-mentioned metadata), (e.g. using description.keyframe, description.startTime, description.endTime, description.text)

2. The second level is for metadata on the story units, obtained using a small subset of extended Dublin Core metadata (we detail this level below)

3. A third level is for metadata on the keyframe (possibly based on clustering processes), that uses Virtual Image (described in detail in the next section)

In particular we will focus in the second level; for this one only the following metadata are necessary:

- Subject: Since story units are created for cataloguing and fruition, this element functions as title and subject at the same time. In fact, while for the video a known title of the work usually exists, for the story units it does not exist; then in the story units we can to indicate the category (as action, dialogue, etc.)
• Description: For this element we use the following extensions:
  – Description.Text
  – Description.Keyframe

• Type: With it we indicate the type of resource between the possible ones for the video streaming (as video, scene, shot, frame, at which we add story unit)

• Relation: This element is important since it implicitly allows inheriting from video the remaining Dublin Core metadata. In fact in the story units (and in the scenes that compose the story units) we use the descriptor Relation.IsPartOf; it joins such entities to “father” video (the video from which we extract scenes and story units). Then we derive the remaining attributes from the “father” video. Moreover for the story units we propose the Relation.HasPart extension, in order to connect story unit with scenes whose it is composed.

It is necessary to focus on the Description.Keyframe elements, which represents story units and then scenes. It is just the beginning point of our content& metadata based cataloguing. Then we can modify the scheme of figure 5.2 as figure 5.6.

5.4 EXPERIMENT AND RESULTS

Extensive experiments shows the resulting video indexing process as shown in figure 5.6.
Fig. 5.6 The content and metadata based video indexing

5.5 CONCLUSION

In the proposed work we are trying to integrate a single video indexing process by two different methods, 1) the meta-based method based on the use of Dublin Core extensions for video streaming, and 2) the content-based method, through the use of virtual image. Fig.5.6 shows the resulting video indexing process.