Proc. Int. Conf. on Dublin Core and Metadata for e-Communities 2002: 97-103
© Firenze University Press

The Virtual Image in Streaming Video Indexing

Piera Palma, Luca Petraglia, Gennaro Petraglia
Dipartimento di Matematica e Informatica - University of Salerno (Italy)
I-84081 Baronissi, Salerno, Italy
petragli@unisa.it
tel +39 089 965248 fax +39 089 965438

Abstract

Multimedia technology has been applied to many types of applications and the great amount of multimedia data need to be indexed. Especially the usage of digital video data is very popular today.

In particular video browsing is a necessary activity in many kinds of knowledge. 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. In this paper, we focus on the visual and textual content of video for indexing.

In the former approach we use the Virtual Image and in the latter one we use the Dublin Core Metadata, opportunely extended and multilayered for the video browsing and indexing.

Before to concentrate our attemption on the visual content we will explain main methods to video segmentation and annotation, in order to introduce the steps for video keyfeature extraction and video description generation.

Keywords: *Video and Image Indexing, Video Browsing, Keyframe, DC Metadata , Virtual Image.*

1. Introduction

Digital video is becoming the rising tide of multimedia. 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, we usually described and annotated video content 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:

- *Structural level*, and then we divide videos into frames, shots, clips, episodes or scenes;
- *Content level*, according to cinematographic properties, motion of the camera, audio properties, motion of a character/object, scenes and stories within a video, etc.

This paper is organized as follows. In section 2 we describe the two levels of video analysis mentioned above. In section 3 we introduce the criteria of choice for metadata to video indexing and how we apply these metadata to video segments used in our processes of video indexing. In section 4 we describe the Virtual Image and in section 5 we say *why* we use it to video indexing and *how* this content based method can manage also the metadata. In section 6 we make our conclusion on the work.

2. Video Segmentation and Video Extraction/Annotation

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, motions in a video segment. We briefly describe both below. We define segmentation 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 and 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:

- 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;
- 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).

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 [13], 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. 1.

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.

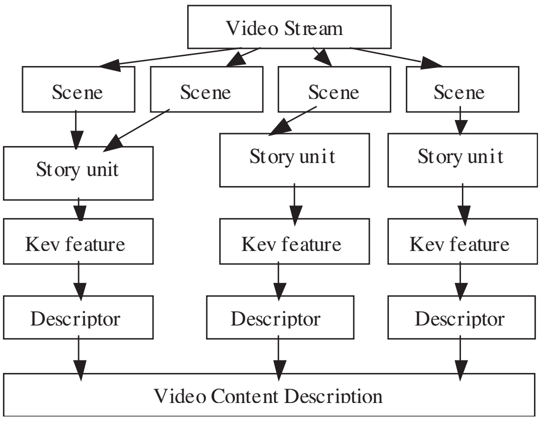

Fig 1. The description generation

Indexes on objects are usually extracted from the keyframe (as mentioned above); the keyfeatures (informations 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 keyfeatures. 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 etablishes specific relationship among objects through their combination and temporal ordering of their actions.

*Icon*s 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 relationships in a sequence is the *iconic annotation by example*, where visual examples are built; these visual examples represent the content of a video segment that will be parsed into a simbolic 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. From these authors we mention Arndt and Chang [1] and Del Bimbo et al. [8]. Arndt and Chang have suggested 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*, that 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 mechanim is particularly used in the querying phase).

3. Metadata in the video indexing process

Currently 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.1, where two video segmentation levels surface: the first level is the *scene*; the second one is the *story unit*.

Definition A *story unit* is the aggregation of many scenes logically connected. It differs from the 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 phisically stored, but it need to be characterized in the system catalog of OODB. Consequently it will provide a greatest amount of informations without futher 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;
- The indexing and querying processes use search engines based on metadata.

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), opportunely extended as specified in J. Hunter's proposal [24] (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 extentions:
  – 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 to ***inherit*** 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* extention, in order to connect story unit with scenes whose it is composed

It is necessary to focus on the ***Description. Keyframe*** element, that 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 Fig. 2 as follows:

st the beginning point of our *content&metadata*
'd cataloguing. Then we can modify the scheme of
as follows:

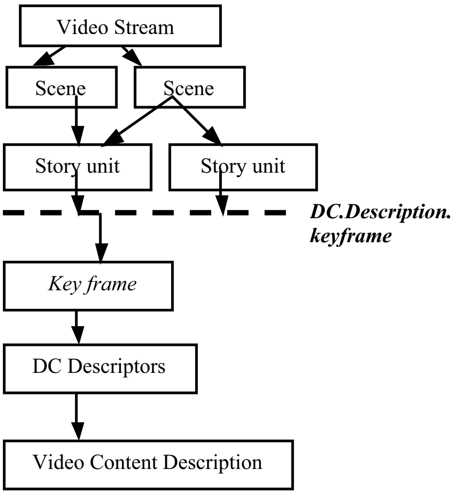

Fig 2. The *metadata-based* video indexing

4. The Virtual Image

From original point of view the Virtual Image [17] describes its corresponding real image in terms of objects and spatial relationships and preserves the spatial knowledge embedded in the real image.

Formally it is defined as a pair (Ob, Rel) where :

- Ob = {ob_1, ... , ob_n} is a set of objects
- Rel = {Rel_x, Rel_y} is a couple of sets of binary spatial relationships on Ob, in particular Rel_x (resp. Rel_y) contains disjoint subsets of Ob x Ob that

express spatial relationships "<" , "|","=", "[", "]", "/", "%", between object pairs of *im* (the real image) on *x* axis (resp. *y* axis)

For simplicity we use the notation $ob_i \ \gamma \ ob_j$ to indicate that the pair (ob_i, ob_j) belongs to the relation γ, where ob_i, $ob_j \in$ Ob and $\gamma \in \{>, | , = , [,] , / , \%\}$. A triple like $ob_i \ \gamma \ ob_j$ is called an *atomic relation* in the following. We also say that atomic relation $ob_i \ \gamma \ ob_j$ *belongs to* Rel_x (resp. Rel_y) if the spatial relation holding between ob_i and ob_j along the x - projection (resp. y – projection) is γ. We can regard both Rel_x and Rel_y simply as sets of atomic relations. In the figure below we show possible spatial relations:

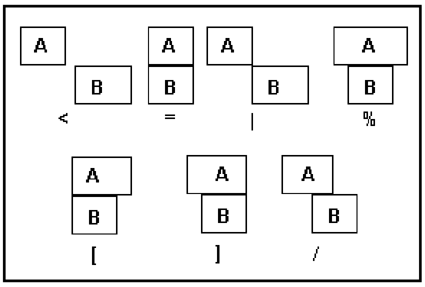

Fig 3. Example of possible spatial relations betweeen A and B

The Atomic Relation Extraction Method (AREM algorithm) derives Virtual Image from a given real image through the following steps:

Step 1: Let Rel_x (resp. Rel_y) be empty set;

Step 2: Scan the image along the x-direction (resp. y-direction) to compute the values begin (A) and end (A) for every A∈ Ob;

Step 3: For each couple (A,B) ∈ Ob, add to Rel_x (resp. Rel_y) the relation obtained by the following case-statement:

```
Case:
 end(A)<begin(B)                              : A<B
 end(B)<begin(A)                              : B<A
 begin(A)=begin(B) and end(A)=end(B)          : A=B
 end(A)=begin(B)                              : A|B
 end(B)=begin(A)                              : B|A
 begin(A)<begin(B) and end(A)>end(B)          : A%B
 begin(B)<begin(A) and end(B)>end(A)          : B%A
 begin(A)=begin(B) and end(A)>end(B)          : A[B
 begin(A)=begin(B) and end(A)<end(B)          : B[A
 begin(A)<begin(B) and end(A)=end(B)          : A]B
 begin(A)>begin(B) and end(A)=end(B)          : B]A
 begin(A)<begin(B) < end(A)<end(B)             : A/B
 begin(B)<begin(A) < end(B)<end(A)            : B/A
end Case
```

5. Virtual Image as bivalent interface between icons and metadata

In section 3 we introduced the story unit keyframe concept: for us it constitutes the joining element between metadata-based indexing and content–based one. Such joining is realized expanding the Virtual Image. This concept has been introduced to keyframe characterization in video segmentation and video annotation [12].

As we saw in previous section, in its original form the Virtual Image is a string of spatial relationships between objects obtained through AREM algorithm. We extend this structure providing it of Dublin Core metadata [25]. As above mentioned,] the Virtual Image is proposed as a video indexing way through the use of keyframe indexing. Then it is possibile to characterize obj not as real elements of the objects existing in the keyframe, but as elements formed by iconic image of element and metadata associated. In such way, from one side keyframe is a representative element of a video segment (shot, episodes, scenes or story unit), from the other one it is possible to index it with Virtual Image. Since in [17] the effectiveness of Virtual Image has been demonstrated in content based image indexing, we focus on the importance of introduction of metadata in the Virtual Image and in its obj elements.

In [11] there is a DDL defined using SQL-like terms for the Virtual Image and then including the metadata. We extend this concept to the streaming video providing Virtual Image of metadata at two levels. The higher level includes metadata of real image (in our case is the keyframe) from which we derived Virtual Image. Instead lower level includes metadata for the *n* objects whose Virtual Image is composed; then the ob_j will be stored in a database with the AREM String and the relative metadata; such objects will be used for querying and retrieval. Then from one side Virtual Image is able to make content-based indexing (through the string of spatial relationships obtained by AREM method), from the other one it is able to index through Dublin Core metadata. Obviously it is possible to use the two methods together because Virtual Image includes both. Actually we are studying this point with many streaming video.

Then Virtual Image realizes a **biunique correspondence** between iconic content (needed by user i

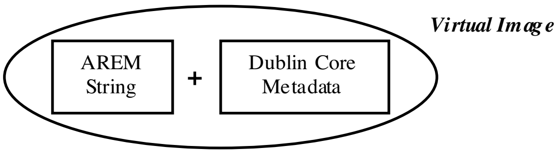

Fig 4. The New Virtual Image Universe

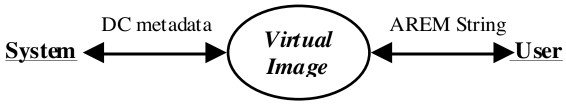

Fig 5. Virtual Image as interface between content and metadata

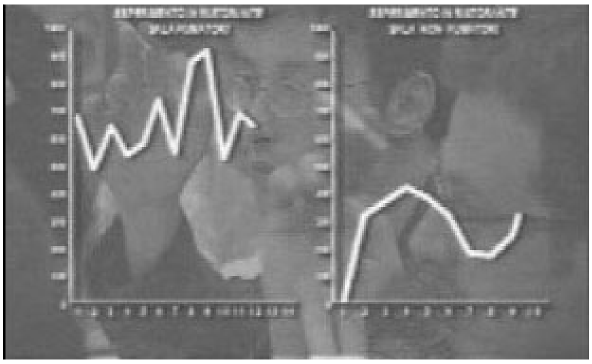

Fig 6: Keyframe example

the querying phase) and metadata relative to it (used by system); we schematize this concept in Fig. 5.

Let us as example a keyframe extracted from a video documentary (Fig. 6); from it we can extract the significant objects. Then we provide these objects of the Minimum Bounding Rectangle (MBR); we call the obtained image *symbolic image* (Fig. 7), that will be the input of the AREM algorithm.

Finally we show the Virtual Image resulting from the application of the AREM algorithm and the description through metadata in the Table 1.

Table 1. Virtual Image and Metadata for the example keyframe

Keyframe	
DC.Description.text	Graphs
DC.Relation.IsPartOf	Scene D
Objects	A,B,C,D
AREM.X	A<B, A%C, A<D, B%D, C<B, C<D
AREM.Y	A=B, C/A, C/B, C=D, D/A, D/B

As we can see in the table above we included in the Virtual Image of the keyframe the metadata and in particular:

Description.text, that is a little description of the keyframe (subjective information);

Relation.IsPartOf, that relates keyframe with the video segment (scene) or video segment aggregate (story unit) from which it is extracted (objective information).

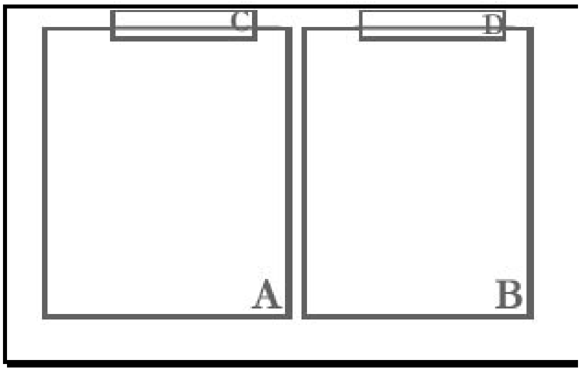

Fig 7. Symbolic Image of the Keyframe

Since the only metadata on the content of the keyframe (that is description.text) is a subjective information (it depends on the person assigned to database population), Virtual Image provides a more objective description of the keyframe content.

In the example keyframe there are two graphs that we havo to compare: in this case it is very important the way in which the graphs are disposed, and consequently spatial relations (between the objects) of the keyframe are important, then Virtual Image extended to metadata provides a complete description of it.

5. Conclusion and Future Works

In this paper we looked to integrate in a single video indexing process two different kinds of approach: the metadata based approach, based on the use of Dublin Core extentions for video streaming, and the content based one, through the use of Virtual Image. We can schematize the resulting video indexing process in Fig.8.

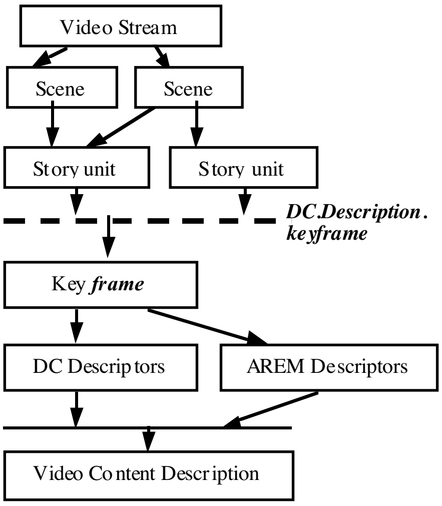

Fig 8. The *content & metadata based* video indexing

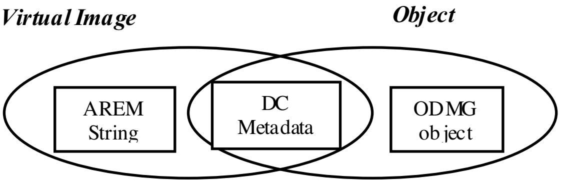

Fig 9. Virtual Image as future Integration method

In addition to this mechanism we are designing an integration system of the whole video indexing mechanism with an *Video Digital Repository*, based not only on an efficient video storing, scenes and frames, but also based on attributes directly derived from ODMG 3.0 standard. In such way Virtual Image will be a more important instrument for its ability to integrate the standards that are actually extending. We show this idea in Fig. 9.

References

[1] Arndt, T. e S.K. Chang, 1989. "Image sequence compression by iconic indexing". *IEEE VL Workshop on Visual Languages*, Roma, Italy.

[2] Chang S.K., Q.Y. Shi, e C.W. Yan. "Iconic indexing by 2D-String". *IEEE Transactions on Pattern Analysis and Machine*, 9(3).

[3] Cattell R.G.G. et al. 2000. "The object data standard: ODMG 3.0" by Cattell R.G.G. and Barry D.K. Morgan Kaufmann Publisher Inc.

[4] Chang, S.K. et al. 1996. "Symbolic Projection for Image Information Retrieval and Spatial reasoning". *Signal Processing and its applications* by R. Green D. Gray and E.J. Powers Academic Press, pag. 68-70.

[5] Corridoni, J.M., A.Del Bimbo, D. Lucarella, H. Wenxue, 1996. "Multiperspective Navigation of Movies". *Journal of Visual Languages and Computing*.

[6] Davis, M., 1993. "Media Streams, an iconic visual language for video annotation". *Proceedings IEEE VL 93 Workshop on Visual Languages*. Bergen, Norway.

[7] Del Bimbo, A., 2000. "Visual Information Retrieval". Morgan Kaufmann Publishers, Inc.

[8] Del Bimbo, A., E.Vicario e D.Zingoni, 1995. "Symbolic description and visual querying of image sequences using spatio temporal logic". *IEEE Transactions on Knowledge and Data Engineering*.

[9] Flickner, M., H. Sawney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, 1995. "Query by *Image and Video Content: the QBIC System". IEEE Computer*.

[10] Freksa C., 1992. "Temporal Reasoning based on Semi-Intervals". *Artificial Intelligence*, 54 (1,2).

[11] Landi, R., L. Petraglia, G. Petraglia, 2002. "The Data Definition Language for the Virtual Image". *Proceeding of the 20th IASTED International Conference on Applied Informatics*, Feb 18-21, Innsbruck, Austria.

[12] Lee, H., A.F. Smeaton, C. Berrut, N. Murphy, S. Marlow and N.E. O'Connor, 2000. "Implementation and Analisys of Several Keyframe-Based Browsing Interfaces to Digital Video". *Research and Advanced Technology for Digital Libraries, 4th European Conference, Lisbon, Portugal. ECDL 2000.*

[13] Lee, S.Y., S.T. Lee e D.Y. Chen, 2000. "Automatic Video Summary and Description". In *Advances in Visual Information Systems*, VISUAL2000, 4th International Conference, Lyon.

[14] Lee, S.Y. and F.J. Hsu. "Spatial reasoning and similarity retrieval of images using 2D C-String Knowledge Representation". In *Pattern Recognition*, 25, 1992, 305-318.

[15] Nabil M., A.H.H. Ngu and J. Shephard, 1996. "Picture Similarity Retrieval using 2D Projection Interval Representation". *IEEE Transaction Knowledge and Data Engineering*, 8(4).

[16] Palma P. and G. Petraglia, 2002. "Video Indexing: Keyframe or Visual Icon?" In *IASTED International Conference* (submitted).

[17] Petraglia, G., M. Sebillo, M. Tucci, and G. Tortora, 2001. "Virtual images for similarity retrieval in Image databases". *IEEE Transaction on Knowledge and Data Engineering* , 13 (6).

[18] Petraglia, G., M. Sebillo, 1997. "The Virtual Image as object-relational database". *Proceeding of the 15th IASTED International Conference on Applied Informatics*, Feb 18-20, Innsbruck Austria ed M.H. Hamza IASTED Acta press, pp. 41-44.

[19] Petraglia, L., 2001. "A Federated Multidatabase System for Digital Repository on WAN". Thesis.

[20] Sawney H.S., S. Ayer e M. Gorkani, 1995. "Dominant and multiple motion for video representation". *Proceedings International Conference on Image Analysis and Processing*.

[21] Smoliar S., and H.J. Zhang, 1994. "Content-Based Video Indexing and Retrieval". In *IEEE Multimedia*.

[22] Vendrig G. and M. Worring, 2000. “Feature Driven Visualization of Video Content for Interactive Indexing”. In Advances in Visual Information Systems, VISUAL2000, 4th International Conference, Lyon.

[23] http://archive.dstc.edu.au/RDU/staff/jane-hunter/ECDL3/ paper.html

[24] http://archive.dstc.edu.au/RDU/staff/jane-hunter/ECDL2/final.html

[25] www.dublincore.org