Prototypes selection with context based intra-class clustering for video annotation with MPEG-7 features

Costantino Grana, Roberto Vezzani, Rita Cucchiara
Department of Information Engineering, University of Modena and Reggio Emilia, Italy
{grana.costantino, vezzani.roberto, cucchiara.rita}@unimore.it

Abstract

In this work, we analyze the effectiveness of perceptual features to automatically annotate video clips in domain-specific video digital libraries. Typically, automatic annotation is provided by computing the clip similarity with respect to pre-annotated examples, which constitute the knowledge base, in accordance with a given ontology or a classification scheme. The amount of training clips is normally very large, in order to capture the structure and the variability of each class. Instead of using the whole knowledge base, we propose to automatically extract some prototypes, or visual concepts, for each class. They are generated after a process of Complete Link clustering based on perceptual features with an automatic selection of the number of clusters. Context based information are used in an intra-class clustering framework to provide selection of more discriminative clips. Reducing the number of stored samples makes the matching process faster and lessen the storage requirements of the visual concepts. Clips are annotated following the MPEG-7 standard directives to provide easier portability. Results are provided on videos taken from sports and news digital libraries.

Categories and Subject Descriptors

H3.7 [**Information Storage and Retrieval**]: Digital Libraries – *collection, standards, system issues.*

General Terms

Algorithms, Documentation, Performance, Standardization.

Keywords

MPEG-7, video annotation, sports video, context based clustering

1 Introduction

In the last decade, a significant increase of the availability of devices able to acquire and store digital videos and the introduction of broadband connections has given a strong impulse to the study and development of video digital libraries management systems. In particular, a growing need is the ability to search for videos basing on their content instead of relying on manually provided metadata. The diffusion of such systems has been strongly limited by the difficulty to generalize results of visual and aural automated processing techniques obtained on tuned test data sets. On the other hand, general internet users are very inexperienced in search, so the media search technologies for the mass have to be very simple, intuitive, and easy to use (Jaimes *et alii* 2005).

From the technical point of view, another question is whether we should go for domain-dependent features or for more general ones, defined at perceptual level only. This last choice could be probably less effective than ad-hoc defined features, but is potentially applicable to wider scenarios. Moreover, the required level of detail goes often beyond the video granularity, in the sense that the user is looking for smaller subsequences, i.e. clips which could be shots or even finer temporal segments.

Examples of automatic semantic annotation systems have been presented recently, most of them in the application domain of news and sports video. Most of the proposals deal with a specific context making use of ad-hoc features. In the paper by Bertini *et alii* 2005, the playfield area, the number and the placement of players on the play field, and motion cues are used to distinguish soccer highlights into subclasses. Differently, a first approach trying to apply general features is described by Chang *et alii* 1998. Employing color, texture, motion, and shape, visual queries by sketches are provided, supporting automatic object based indexing and spatiotemporal queries.

In this paper, we propose a general framework which allows to automatically annotate video clips by comparing their similarity to a domain specific set of prototypes. In particular, we focus on providing a flexible system directly applicable to different contexts and a standardized output by means of the MPEG-7 tools. To this aim, the clip characterizing features, the final video annotation, and the storage of the reference video objects and classes are realized using this standard.

Starting from a large set of manually annotated clips, according with a classification scheme, the system exploits the potential perceptual regularity and generates a set of prototypes, or visual concepts, by means of a intra-class clustering procedure. Then, only the prototypes are stored as suitable specialization concepts of the defined classes. The adoption of the limited set of prototypes instead of the whole set of examples reduces the required storage space and allows the generation and the sharing of a context classifier on the Web. Thanks to the massive use of the MPEG-7 standard, a remote system could then perform its own annotation of videos using these context classifiers.

As most of the papers, we refer to already subdivided clips. The automatic subdivision of videos into clips is a widely faced problem, and several solutions are available. Our system uses the approach described in Grana *et alii* 2005, followed by a fuzzy c-means frame clustering to provide clips at sub-shot granularity.

The paper is organized as follows: in Section 2 a similarity measure between clips based on standard low level features is described. Based on it, a nearest neighbor automatic annotation is presented in Section 3 together with some details about the use of MPEG-7. Section 4 depicts the prototype creation algorithm, the proposed index for automatic level selection, and a context based dissimilarity measure. Results over sports and news videos are reported in Section 5.

2 Similarity of video clips

The problem of clip similarity can be seen as a generalization of the problem of image similarity: as for images, each clip may be described by a set of visual features, such as color, shape and motion. These are grouped in a feature vector:

$$\mathbf{V}_i = \left[F_i^1, F_i^2, \dots, F_i^N \right] \tag{1}$$

where *i* is the frame number, *N* is number of features and F_i^j is the *j*-th feature computed at frame *i*. However, extracting a feature vector at each frame can lead to some problems during the similarity computation between clips, since they may have different lengths; at the same time keeping a single feature vector for the whole clip cannot be representative enough, because it does not take into account the features temporal variability. Here, a fixed number *M* of feature vectors is used for each clip, computed on *M* frames sampled at uniform intervals within the clip. In our experiments, a good tradeoff between efficacy and computational load suggests the use of $M = 5$ for clips of averaging 100 frames. To provide a general purpose system, we avoid to select context dependent features, relaying on broad range properties of the clips. To allow easier interoperability and feature reuse, we tried to select features which comply with the MPEG-7 standard (ISO/IEC Std. 15 938-3:2003). In particular the following three features are employed:

1. Scalable color: a color histogram, with 16 values in H and 4 values in each S and V (256 bins in total).
2. Color layout: to account for the spatial distribution of the colors, an 8x8 grid is superimposed to the picture and the average YCbCr values are computed for each area.
3. Parametric motion: making use of the MPEG motion vectors, the translational motion of each quarter of frame is estimated.

Thus, the distance between two clips S_u and S_v is defined as

$$d\left(S_u, S_v\right) = \frac{1}{M} \sum_{i=1}^{M} \left\| \mathbf{k}^{\mathrm{T}} \left(\mathbf{V}_{u_i} - \mathbf{V}_{v_i} \right) \right\| = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N} k_j \left\| F_{u_i}^j - F_{v_i}^j \right\|, \tag{2}$$

where, $\mathbf{k} = \left(k_1, \dots, k_N\right)$ is a weight vector and u_i and v_i are the frame numbers of the i-th subsampled frames of S_u and S_v respectively. The weights $k_j \in \left[0,1\right]$ provide dimensional coherence among the different features and at the same time they allow to change their relative significance.

3 Automatic annotation

Given a domain-specific video digital library, we assume that it is possible to partition the clips of videos of that context into a set of *L* classes $\mathbf{C} = \left(C_1, \dots, C_L\right)$, which describe different contents or camera views .Given a large set of training clips, we implemented an interactive user-friendly interface (Grana *et alii* 2006) to quickly assign each of them to a specific class C_k and then employ it for automatic annotation purposes. In Fig. 5, a screenshot of the developed annotation tool is shown. An unknown clip can be classified using a nearest neighbor approach and the similarity measure defined above. The weights k_i of Eq. 2 may be tuned to optimize the classification

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<Mpeg7 [...]>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationScheme uri="urn:mpeg:mpeg7:cs:F1_race">
      <Term termID="Graphics" Definition=""/>
      <Term termID="AerialView" Definition=""/>
      <Term termID="CameraCar.Front.Top" Definition=""/>
      <Term termID="People" Definition=""/>
      <Term termID="FarView" Definition=""/>
      [...]
    </ClassificationScheme>
  </Description>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="CollectionModelType">
      <Label href="urn:mpeg:mpeg7:cs:F1_race:People"/>
      <Collection xsi:type="ContentCollectionType" id="prototype0">
        <Content xsi:type="VideoType">
          <Video><MediaLocator><MediaUri>Clip1_2323.avi</MediaUri>
          </MediaLocator></Video></Content>
        <ContentCollection><Content xsi:type="ImageType">
            <VisualFeature xsi:type="ScalableColorType" numOfCoeff="256">
              <Coeff>101376 -51 122 -100 -91694 -69 -620 -185 [...]</Coeff>
            </VisualFeature>
            <VisualFeature xsi:type="ColorLayoutType">
              <YDCCoeff>31</YDCCoeff><CbDCCoeff>20</CbDCCoeff>
              <CrDCCoeff>20</CrDCCoeff>
              <YACCoeff63>13 15 17 13 17 16 18 19 [...]</YACCoeff63>
              <CbACCoeff63>20 17 15 17 15 15 16 16 [...]</CbACCoeff63>
              <CrACCoeff63>7 15 18 15 16 20 15 16 [...]</CrACCoeff63>
            </VisualFeature>
            <GridLayoutDescriptors numOfPartX="2" numOfPartY="2">
              <Descriptor xsi:type="ParametricMotionType" motionModel="translational">
                <CoordDef originX="0" originY="0"/>
                <Parameters>2.828283 -0.404040</Parameters>
              </Descriptor>
              [...]
            </GridLayoutDescriptors>
          </Content></ContentCollection></Collection></Model>
    [...]
  </Description>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics id="ColorLayout">
      <Property><Name>0.40000</Name></Property>
    </Semantics>
    [...]
  </Description>
</Mpeg7>
```

Fig. 1. Example of an MPEG-7 description of a Pictorially Enriched Ontology.

results on a training video, by searching the set which provides the maximum number of correct class assignments. This process is done once for all during the context classifier design, so the optimization phase is not time constrained and an exhaustive search can be employed.

Since the MPEG-7 standard can naturally include pictorial elements such as objects, key-frames, clips and visual descriptors, we used it to store the classification data by means of the double description provided by the *ClassificationSchemeDescriptionType* Descriptor Schema (DS) combined with a *ModelDescriptionType* DS which includes a *Collection ModelType* DS. The classification scheme allows the definition of a taxonomy or thesaurus of terms which can be organized by means of simple term relations. The collection model is instead an *AnalyticModel,* and therefore it describes the association of labels or semantics with collections of multimedia content. The collection model contains a *Content CollectionType* DS with a set of visual elements which refer to the model being described. In particular, we linked the selected clips and a representation of their features by means of the *ScalableColor* Descriptor (D), *ColorLayout* D and the *ParametricMotion* D. In Fig. 1, an example of an MPEG-7 description of a context classifier is provided.

An advantage of adopting a MPEG-7 framework is that other systems may employ the same data enabling interoperability and easier integration.

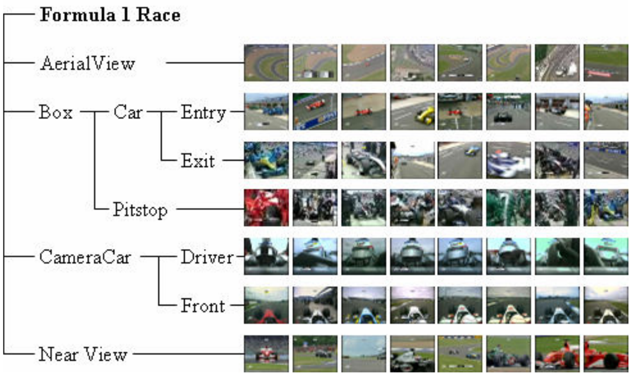


Fig. 2. Example of a few prototypes for the F1 race context. Of the 21 classes available, only 7 classes and 8 prototypes per class are shown.

4 Prototypes creation

As stated above, we adopt a nearest neighbor approach to classify each clip of the test sequences. Increasing the number of training clips the classification performance consequently improves, since a finer partitioning of the feature space is obtained. Unfortunately, in such a manner the space required to store the data, the corresponding transmission time -if needed-, and the computational cost for nearest neighbor selection increase.

4.1 Intra-class clustering

Since not all the clips are equally important to obtain the final classification due to perceptual redundancy in specific domains, we employ a hierarchical clustering method, based on *Complete Link* (Jain and Dubes 1988), to reduce the number of clip of each class, keeping only some representative prototypes, which capture the most significant aspects of a set of clips. This technique guarantees that each clip must be similar to every other in the cluster and any other clip outside the cluster has dissimilarity greater than the maximum distance between cluster elements. For this clustering method a dissimilarity measure between two clusters W_i and W_j is defined as

$$D(W_i,W_j) = \max_{S_x \in W_i, S_y \in W_j} d\left(S_x,S_y\right). \tag{3}$$

where *d* is computed as in Eq. 2. The algorithm proceeds as follows:

1. For each class C_k do steps from 2 to 5:
2. Initially each cluster contains a single clip. Let us call E_n the set of clusters at level *n*, and initialize it to $E_{P_k} = \left\{\{S_1\},\{S_2\},\ldots,\{S_{P_k}\}\right\}$, with $P_k = card\left(C_k\right)$.
3. The least dissimilar pair of clusters, $W_i,W_j \in E_n$, is found according to Eq. 3, i.e. $D\left(W_i,W_j\right) \le D\left(A,B\right)\ \forall A,B \in E_n$.
4. W_i and W_j are merged into the same cluster and E_{n+1} is accordingly updated.
5. If everything is merged into a single cluster or a stop condition is met, the algorithm goes to the next class, otherwise it resumes from step 2.

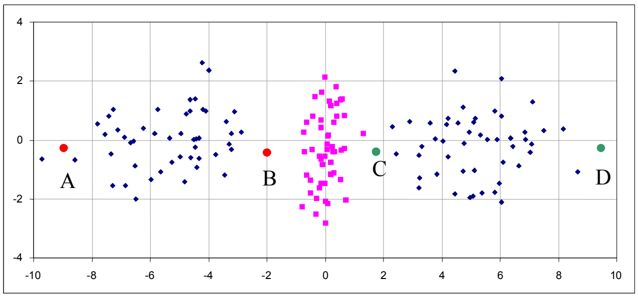
For each class, this algorithm produces a hierarchy of clips partitions with P_k levels, where level 1 is the final step where everything is merged in a single cluster. To implement the algorithm, a proximity matrix was used: initially, it contains the distances between each pair of clips. At each step, the matrix is updated by deleting rows and columns corresponding to the selected clusters and adding a new row and column corresponding to the merged cluster. The values in the new row/column are the maximum of the values in the previous ones.

4.2 Automatic clustering level selection

Instead of a manual selection of the desired clustering level, or a threshold guided one, an automatic selection strategy is proposed. Such a rule has to be based on cluster topology concerns, being a trade-of between data representation and small number of clusters, and it is not possible to choose the *right* one. In literature different proposals are presented, such as the Dunn's Separation Index (Dunn 1973), but the corresponding results on our data sets were not satisfactory. Better results (in terms of a subjective evaluation) have been obtained with the following approach. Let us define the cluster diameter and the cluster distance as

Table 1. Results of Self Annotation on three different videos.

Video	# Frames	# Clips	Correct
Ski	178.181	1212	90%
Bob	50.835	1422	92%
F1	215.638	2339	82%



	AB	BC	CD
distance	**6.9**	**3.7**	**7.7**
dissimilarity	**849**	**2135**	**1050**

Fig. 3. Example of the effect of the context on the dissimilarity measure.

$$\Delta(W_i) = D(W_i, W_i) \tag{4}$$

$$\delta(W_i, W_j) = \min_{S_x \in W_i, S_y \in W_j} d(S_x, S_y) \tag{5}$$

The *Clustering Score* at level *n* is defined as

$$CS_n = \min(\Delta_1 - \Delta_n, \delta_n) \tag{6}$$

where

$$\begin{aligned} \Delta_n &= \max_{W_i \in E_n} \Delta(W_i) \\ \delta_n &= \min_{W_i, W_j \in E_n, i \neq j} \delta(W_i, W_j) \end{aligned} \tag{7}$$

The selected level is the one which minimizes CS_n. It is possible to observe that Δ_n and δ_n are both monotonically increasing with *n*, thus CS_n has a unique global minimum. Therefore, the clustering algorithm can be stopped when CS_n start to increase without computing the remaining levels. A single prototype can be generated from each cluster, by computing the *M* average feature vectors. The clip which minimizes the distance from the prototype features is associated to it, in order to provide a representative of the visual concept. An example of some F1 classes and visual concepts is provided in Fig. 2.

The automatic selection of prototypes allows a remarkable reduction of examples. We tested in many different contexts, such as the Torino 2006 Olympic Winter Games, soccer matches, Formula 1 races, news, and the automatic clustering selects about 10% to 30% only of the provided clips as visual concepts (some example are reported in Table 1).

4.3 Intra-class clustering with context data

In section 4.1 an intra-class clustering has been presented in order to generate a set of significant prototypes for each class. The choice is guided by how similar the original clips are in the feature space, without considering the elements belonging to the other classes (*context data*). This may lead to a prototype selection which is indeed representative of the class but lacks the properties useful for discrimination purposes. To better understand this concept, an example is reported in Fig. 3, in which 150 random samples are extracted from three different Gaussian distributions. Two of them (blue color) belong to the same class, while the third distribution (purple color) is a different one. With the clustering technique described in the previous section, the generation of the prototypes for the first class does not take into account the second distribution; the automatic index selection could merge the two distributions leading to a single prototype. To cope with this problem, a context based similarity measure is provided as follows. We define an *isolation coefficient* for each clip as:

Table 2. Results of Test 2. NN: Nearest Neighbor classification with the originally selected training clips, CL: classification after prototype creation with classic Complete Link clustering, CBCL: classification after prototype creation with Context-Based Complete Link clustering.

		Ski	**Bob**	**F1**
#Training set		300	300	500
#Test set		912	1122	1839
# of Visual Concepts	NN	300	300	500
	CL	84	126	191
	CBCL	78	122	203
Results on training set	NN	300 (100%)	300 (100%)	500 (100%)
	CL	299 (99.7%)	292 (97.3%)	478 (95.6%)
	CBCL	300 (100%)	285 (95%)	492 (98.4%)
Results on test set	NN	660 (72.4%)	854 (75.8%)	1181 (64.2%)
	CL	654 (71.7%)	846 (75.1%)	1159 (63.0%)
	CBCL	657 (72%)	852 (75.7%)	1209 (65.7%)

$$\gamma(S_u)=\sum_{i=1,i\neq j}^{L}\sum_{S_v\in C_i}\frac{1}{d(S_u,S_v)},S_u\in C_j\,. \tag{8}$$

Then we can introduce a class based *dissimilarity measure* between two clips as:

$$\bar{d}(S_u,S_v)=d(S_u,S_v)\cdot\gamma(S_u)\cdot\gamma(S_v)\,. \tag{9}$$

The *intra-class complete link* clustering is thus enhanced with context data by substituting the clips distance in Eq. 3 and Eq. 5 with this dissimilarity measure.

In Fig. 3 four samples of the blue class have been selected. Even if the central points (B,C) are closer each other than the corresponding colored ones (A and D respectively), the interposed purple distribution largely increases their dissimilarity measure, preventing their merge in a single cluster.

5 Experimental Results

The knowledge representation by means of textual and pictorial concepts is particularly suitable for summarization and fast video access. The described system has been tested on video of different contexts: here we provide experiments to test the effectiveness of these prototypes and we propose a semiautomatic annotation framework, to speedup a metadata creation task.

5.1 Automatic annotation

As a first test a self-annotation task was performed, i.e. we automatically classify the clips of a video using a leave-one-out cross validation (i.e., each clip is classified exploiting all the other clips of the same video). Since this corresponds to counting how many clips belong to the same class of their nearest neighbor, we can check how separable the defined classes are in the selected feature space. For the second test, we checked the generalizing properties of the context classifier by using two different sets for the training and test phases.

In Table 1 the results of the first test on three different videos are reported, confirming that the selected feature

Table 3. Confusion matrix of the first experiment of Table 2.

	1	2	3	4	5	6	7	8	9	10	11
1. AerialView	**53**							47			
2. Box.Car.Entry		**78**	10	8	3						
3. Box.Car.Exit		9	**81**	6	4						
4. Box.PitStop	3	6	17	**57**	6				6	6	
5. Box.Staff.Waiting			12	41	**18**				12	18	
6. InCarCam.Front.Bottom						**100**					
7. InCarCam.Front.Top			1				**97**		1		
8. FarView		2	2					**47**	49		
9. NearView			12					12	**76**		
10. People	10	3	7	10			3		14	**41**	10
11. Advertising	1		1			1		1			**96**

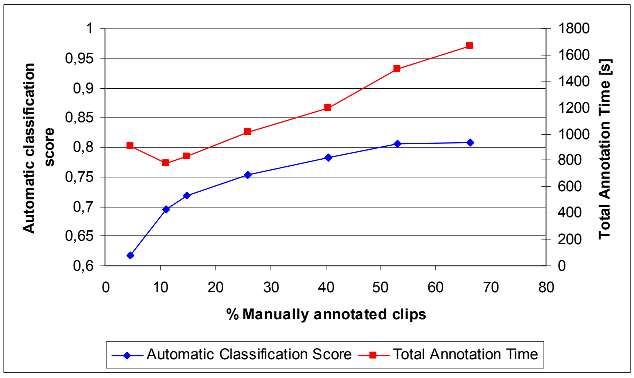


Fig. 4. Semiautomatic annotation framework example. Manually annotating all the clips is slower than correcting the errors of automatic tools.

space is quite effective in the considered contexts. In particular, the ski context presents an higher degree of repetitiveness, so the visual classification performs better than in the F1 one.
To create training sets for different contexts, we used a first part of each video and the rest was used as test. The results of this test are reported in Table 2. All the training clips have been reduced by the prototype creation algorithm and the number of the generated prototypes is about a third of the initial samples. Other experiments on larger training sets have shown higher reduction rates, which also depend on the number of sample per class. The results obtained on the training set show that the clustering process is able to keep the original structure of the data. On the test set, it is possible to see that the use of this approach reach a classification rate around 70% on average, depending on the context. In Table 3 the confusion matrix of the F1 experiment is reported. Looking at Table 3 it is possible to see that the classification provides good results with specific and well constrained classes (e.g. InCarCamera); for more generic classes (e.g. AerialView-FarView), the training data could not be sufficient to describe all the possible samples, so the correspondent results are worse.

5.2 Semi-Automatic Annotation Framework

From previous experiments, it is clear that without context specific features it is not feasible to reach very high automatic classification rates. We believe that 70% correct classification with a generic feature set is a realistic figure. A possible approach to the distribution of annotated videos over internet may be the use of this kind of generic tools followed by manual correction of errors, as it is common for example with every day use of OCR software. To this aim we tested on some winter Olympic Games of Torino 2006 the speedup obtained using our automatic classifier followed by a manual error correction instead of a completely manual annotation. In Fig. 4 the results over a sample video are shown. The score of the automatic classification of the non annotated clips grows with the rise of the manually annotated ones; even though, it is not convenient to annotate to much clips since the increase in correct annotations does not compensate the increased time requirements. In our experiments, the best compromise is reached around 10%.

6 Conclusions

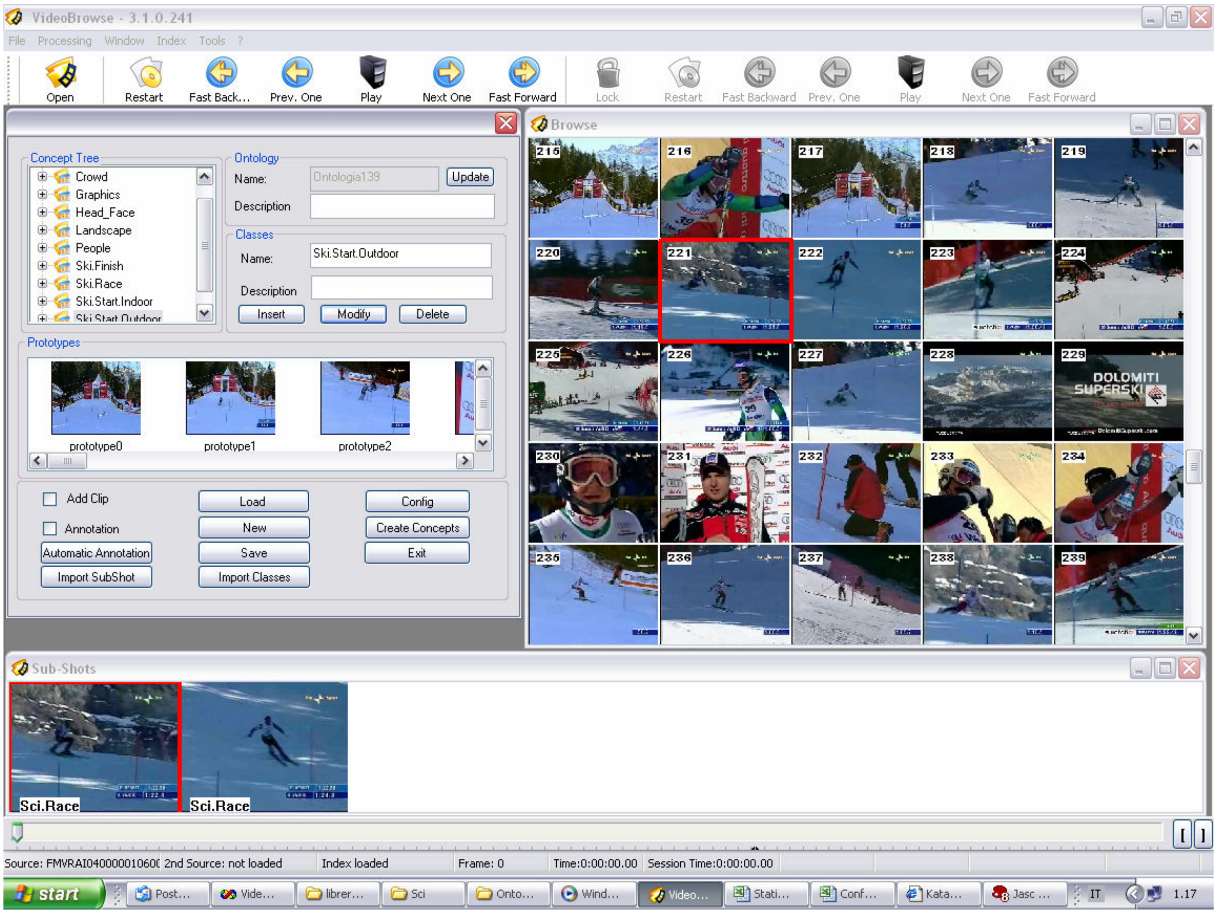
We presented a general purpose system for the automatic annotation of video clips, by means of standardized low level features. A process for reducing space and computational requirements by the creation of prototypes with context based intra-class clustering was described. The classifier data are stored in MPEG-7 compliant format to improve the interoperability and integration of different systems. The annotation has shown satisfactory results without specific feature development. This approach allows a system to behave differently by simply providing a different context, thus expanding its applicability to mixed sources digital libraries.

Acknowledgments

This work is supported by the DELOS NoE on Digital Libraries, as part of the IST Program of the European Commission (Contract G038-507618).

References

Bertini M., Cucchiara R., Del Bimbo A. and Torniai C. 2005. Video Annotation with Pictorially Enriched Ontologies. In *Proceedings of IEEE International Conference on Multimedia and Expo*, Amsterdam, The Netherlands, 1428-1431.



Ontology Name: Ontologia139
Description:

Classes Name: Ski.Start.Outdoor
Description:

Source: FMVRAI04000001060(2nd Source: not loaded Index loaded Frame: 0 Time:0:00:00.00 Session Time:0:00:00.00

Fig. 5. Screenshot of the semi-automatic annotation framework. A ski video is opened. On the right a shots window shows the automatically detected shots, on the lower part a sub-shots windows reports a finer granularity division. On the left, the classification scheme together with the selected prototypes is drawn.

Chang S., Chen W., Meng H.J., Sundaram H. and Zhong, D. 1998. A Fully Automated Content-Based Video Search Engine Supporting Spatiotemporal Queries. *IEEE Transactions on Circuits and System for Video Technology* 8(5): 602-615.

Dunn J.C. 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. In *Journal of Cybernetics* 3(3): 32-57.

Grana C., Tardini G. and Cucchiara R. 2005. MPEG-7 Compliant Shot Detection in Sport Videos. In *Proceedings of the IEEE International Symposium on Multimedia (ISM 2005)*, Irvine (CA), USA, 395-402.

Grana C., Vezzani R., Bulgarelli D., Barbieri F., Cucchiara R., Bertini M., Torniai C. and Del Bimbo A. 2006. PEANO: Pictorial Enriched ANnotation of VideO. In *Proceedings of the 14th ACM international Conference on Multimedia* (Santa Barbara, CA, United States, October 23-27, 2006). 793-794.

ISO/IEC Std. 15 938-3:2003. Information technology - Multimedia content description interface - Part 3: Visual.

Jaimes A., Christel M., Gilles S., Sarukkai R. and Ma W. 2005. Multimedia information retrieval: what is it, and why isn't anyone using it? *In Proceedings of the 7th ACM SIGMM international Workshop on Multimedia information Retrieval* (Hilton, Singapore, November 10 - 11, 2005). MIR '05. ACM Press, New York, NY, 3-8. DOI= http://doi.acm.org/10.1145/1101826.1101829

Jain A.K. and Dubes R.C. 1988. *Algorithms for clustering data*. Englewood Cliffs, NJ: Prentice-Hall.