A Relevance Feedback Approach to Video Genre Retrieval

Ionuţ Mironică[1], Constantin Vertan[1], Bogdan Ionescu[1,2]
[1]*LAPI – University Politehnica of Bucharest, 061071 Romania*
[2]*LISTIC – Université de Savoie, Annecy, 74944 France*
ionut.mironica@gmail.com, constantin.vertan@upb.ro, bionescu@alpha.imag.pub.ro

Abstract

***Abstract*—Content-based retrieval in video databases has become an important task with the availability of large quantities of data in both public and proprietary archives. Most of video systems are based on feature classification, but problems appear because of "semantic gap" between high-level human concepts and the machine-readable low-level visual features. In this paper we adopt a relevance feedback approach (RF) to bridge the semantic gap by progressively collecting feedback from the user, which allows the machine to discover the semantic meanings of objects or events. Experimental tests conducted on more than 91 hours of video footage show an improvement of up to 90% in retrieval accuracy, compared to classic classification-based retrieval.**

Keywords-component; content based video retrieval, relevance feedback, hierarhical clustering

I. Introduction

Many video search engines that are able to explore large multimedia databases have been created in the last 10 years. Major search engines (such as Google and Yahoo) have recently begun to provide content-based video retrieval services (CBVR). The majority of these systems use word descriptions (title, movie synopsis, etc.), but a large amount of extra work is required in order to provide these key words. Other systems generate text surrogates with automatic speech recognition from the audio track of the video [1] or video OCR [2]. This works well when the text content is in some way descriptive of the visual content, or at least the on-screen video is illustrative of the text content. This occurs in video genres like broadcast news or TV documentaries, but not in movies, CCTV, home movies, or other kinds of TV program. Issues regarding this method appear when the user is rather interested in the visual content of the movie than in what there is spoken about.

When text-based video search does not work, systems use another type of search based on key frames similarity. Such matching is usually based on visual similarity between videos and will be useful when a searcher has a candidate query video (or a group of images), or can locate a key frame serving this purpose from the video archive. Other systems try to query the video database, using a set of containing objects obtained by segmentation [3].

In practice, in contemporary video retrieval systems, a combination of text search, key frame matching, and feature annotations are often used together and provide the most useful way to search video when operating collectively [5].

Content-based video retrieval techniques start with extracting low-level features from videos and end up determining similarity between them by computing distances between feature vectors. Most of them focus on extracting color and texture features [4], or region shapes using object segmentation [3].

The main problem is related to the difference between the semantic field in which features are defined and the one the query actually refers to. The general features extracted from the video (histograms, layouts, texture) are low level features that do not match the semantics of an video [4].

The semantic gap characterizes the difference between two descriptions of an object by different linguistic representations, for instance languages or symbols. In computer science, the concept is relevant whenever ordinary human activities, observations, and tasks are transferred into a computational representation [6]. To be more precise, the semantic gap means the difference between ambiguous formulation of contextual knowledge in a powerful language (e.g. natural language) and its reproducible and computational representation (like the feature vector used for video content description).

In order to fill up this semantic gap, two main approaches are possible: one is to index the videos by their semantic contents [13], another, that we will discuss in this paper, is to take advantage directly of the user's expertise (being the "consumer of the product") and thus to adapt the system's response to his needs [6].

The main purpose of this paper is to show improvements of genre-based video retrieval using relevance feedback algorithms. We aim to select the best-suited algorithm by making a comparison of various relevance feedback methods, namely: Rocchio, Robertson-Sparck-Jones, Feature Relevance Estimation, Support Vector Machines and Hierarchical Clustering.

The paper is organized as follows: we describe in Section II the classic content based video-retrieval systems, Section III includes a brief discussion about classical video descriptors. Then, in Section IV we describe relevance feedback methods used in our experiments. Experimental results are discussed in Section V and conclusions in Section VI.

II. Content based video retrieval

The most popular CBVR paradigm is the query by example; Figure 1 schematically synthesizes the architecture of such a system.

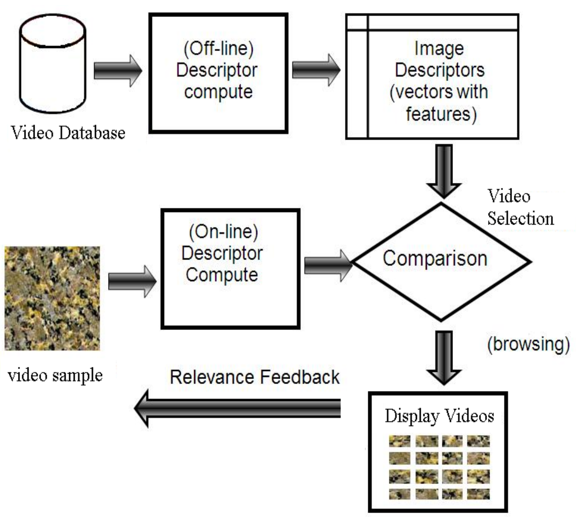

Figure 1. Example of classical query by example CBVR system.

The video database stores all offline computed video descriptors and then the system calculate top k nearest video documents.

In this paper we use three categories of content descriptors: temporal, visual and structural. Temporal descriptors are derived using a classic confirmed approach, thus analyzing the frequency of shot changes.

Color descriptions use statistics of color distribution, elementary hues, color properties and relationship of color [7].

Finally, video structural information describes curve contour geometry, individually and in relation with neighbor contours [8].

III. Content descriptors

A. Temporal descriptors

Temporal descriptors aim at capturing the temporal structure of the movie. Different video genres show specific patterns, e.g. commercials and music tends to have a high visual rhythm and action content, while documentaries show a slower visual change.

Rhythm. To capture the movie's visual changing tempo, first we compute the relative number of shot changes occurring within a time interval T = 5s, denoted ζT (we detect cuts, fades and dissolves). Then, the rhythm is defined as the movie average shot change ratio, thus $E\{\zeta T\}$.

Action. We aim at highlighting two opposite situations: video segments with a high action content (denoted hot action) with $\zeta T > 2.8$, and video segments with low action content with $\zeta T < 0.7$ (thresholds were set based on human observation of different action levels). We compute the hot-action ratio, HA = THA/Tvideo and low-action ratio LA = TLA/Tvideo, where TX represents the total duration of all type X sequences.

Gradual transition ratio. High amounts of gradual transitions are in general related to a specific video contents, therefore we compute: GT = (Tdissolves + Tfade−in +Tfade−out)/Ttotal where TX represents the total duration of all the gradual transitions of type X.

B. Color information

Color information is a powerful descriptor for deriving information about the visual perception of the sequence. Different genres show different color patterns, e.g. animated movies have specific palettes, music videos tend to have darker colors, sports usually show a predominant hue, etc. We extend image-based color descriptors at temporal level and provide a global color content description using color statistics, elementary hues, color properties and relationship of color [7].

Global weighted color histogram is computed as the weighted sum of each shot color histogram, thus:

$$h_{GW}(C) = \sum_{i=0}^{M} \left[\frac{1}{N_i} \sum_{j=0}^{N_i} h^{j}_{shot_i}(c) \right] \frac{T_{shot_i}}{T_{total}} \tag{1}$$

where M is the total number of video shots, N_i is the total number of the retained frames for the shot i (we use temporal sub-sampling), $h_j^{shot_i}$ is the color histogram of the frame j from the shot i, c is a color index from the Webmaster palette (we use color reduction) and T_{shoti} is the length of the shot i. The longer the shot, the more important the contribution of its histogram to the movie's global histogram.

Elementary color histogram. The next feature is the distribution of elementary hues in the sequence, thus:

$$h_E(C_e) = \sum_{c=0}^{215} h_{GW}(c) \mid Name(c_e) \subset Name(c) \qquad (2)$$

where ce is an elementary color from the Webmaster color dictionary (colors are named according to color hue, saturation and intensity) and Name() returns a color's name from the palette dictionary.

Color properties. Considering the color naming dictionary provided with the Webmaster's palette, we define several color ratios. For instance, light color ratio, Plight, reflects the amount of bright colors in the movie, thus

$$h_E(C_e) = \sum_{c=0}^{215} h_{GW}(c) \mid W_{light} \subset Name(c) \qquad (3)$$

where c is a color with the property that its name contains one of the words defining brightness, i.e. Wlight ∊ {"light", "pale", "white"}. Using the same reasoning and keywords specific to each property, we define dark color ratio (Pdark), hard saturated color ratio (Phard), weak saturated color ratio (Pweak), warm color ratio (Pwarm) and cold color ratio (Pcold) [5]. Additionally, we 0capture movie color wealth with two parameters: color variation, Pvar, which accounts for the amount of significant different colors and color diversity, Pdiv, defined as the amount of significant different color hues.

Color relationship. Finally, we compute Padj, the amount of similar perceptual colors in the movie and Pcompl, the amount of opposite perceptual color pairs.

C.Structural Information

Natural objects in a movie scene are often characterized by curved contours. We use the method proposed in [8] to identify those aspects. Image-level variations of structural properties are captured with histograms, and averaged for the entire movie to form the structure signature of the movie.

Contour properties. First we compute some basic parameters, such as contour orientation (o) and length (l). Then a local/global space with multiple curvature signatures is created similar to the scale space for a contour. From this space we derive the following geometric parameters: arc (a) or alternating (x), whereby the values are scalar and express the strength of these aspects, the curvature parameter (b) that expresses the circularity and amplitude of the arc/alternating contour, the edginess parameter (e) that expresses the sharpness of a curve (L feature or bow) and the symmetry parameter (s) that expresses the evenness of the contour [8].

Contour relations. For each contour, three neighboring segments are searched: one for each endpoint, and one (or two) for its center point that forms a potential pair of parallel segments. Selected pairs are then geometrically described by: the angular direction of the pair (γ), the distance between the proximal contour endpoints (dc), the distance between the distal contour endpoints (do), the distance between the center (middle) point of each segment (dm), the average segment length (l), the symmetry of the two segments (y), the degree of curvature of each segment (b1, b2, which are computed on the curvature values of the 2 segments) and three structural biases that express to what degree the pair alignment is a L feature (ŠL), T feature (ŠT) or a 'closed' feature (Š(), two curved segments facing each other) [8].

IV. RELEVANCE FEEDBACK IN VIDEO SYSTEMS

Traditional CBVR systems do not achieve high performance on general video databases mainly due to several specific problems, the two most important being:

- the difference between the high level features and the low level features, known as the semantic gap. In few cases the assumption that high-level feature concepts have mapped to low-level concepts is correct (e.g. yellow pears have their own color and shape description), but in most cases this is not true (complicated scenes, object with different features);
- the human perception which makes that humans can perceive the same visual content in many, often different circumstances.

Since human perception of image similarity is both subjective and task-dependent, the main method to reduce the semantic gap is the use of relevance feedback (RF). Relevance feedback is an essential component of a CBVR system and means the immediate and explicit assessment of the appropriateness of the original query results by the user. In the following we shall discuss five relevance feedback approaches, thus: Rocchio, Robertson-Sparck-Jones, Feature Relevance Estimation, Support Vector Machine and Hierarchical Clustering.

A. Rocchio's algorithm

One of the earliest and most successful relevance feedback algorithms is the Rocchio algorithm. The Rocchio algorithm uses a set R of relevant documents (containing |R| documents) and a set N of non-relevant documents (containing |N| documents), selected in the user relevance feedback phase, and updates the query features according to the following equation:

$$Q' = \alpha Q + \frac{\beta}{|R|}\sum_{R_i \in R} R_i - \frac{\gamma}{|N|}\sum_{N_i \in N} N_i \tag{4}$$

where the new query Q' is obtained by adjusting the position of the original query Q in the feature space, according to the positive and negative examples and their associated importance factors (importance factor of positive feedback, β, importance factor of negative feedback, γ, and importance of the original query, α). All importance factors are within the [0, 1] range. Figure 2 presents an intuitive graphical representation of the Rocchio relevance feedback principle.

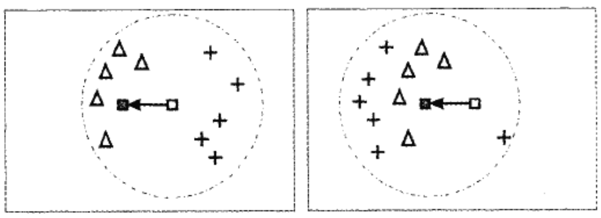

Figure 2: Schematic ilustration of the Rocchio algorithm, as proposed in [17] (Δ - documents marked as relevant, □ - query, + other returned documents).

B. Feature relevance estimation

The feature relevance estimation (RFE) approach assumes, for a given query, that according to the users' subjective judgment, some specific features may be more important than other features [6]. Every feature will have an importance weight that will be computed as Wi = 1/σ, where σ denotes the variance of relevant retrievals, so features with bigger variance have low importance than elements with low variations. The initial weights are equal to 1 and get updated as the user provides the feedback. After applying the relevance feedback, the distance between any two videos becomes a weighted Euclidian distance within their associated feature vectors X and Y:

$$Dist(X,Y) = \sqrt{\sum_{i=1}^{d} W_i (X_i - Y_i)^2 \Big/ \sum_{i=1}^{d} W_i} \tag{5}$$

The modification of the weights associated to the individual features describing the video content means that, in the feature space, the shape of the query selection can be modified from the original sphere to an ellipsoid, as suggested in the example from Figure 3.

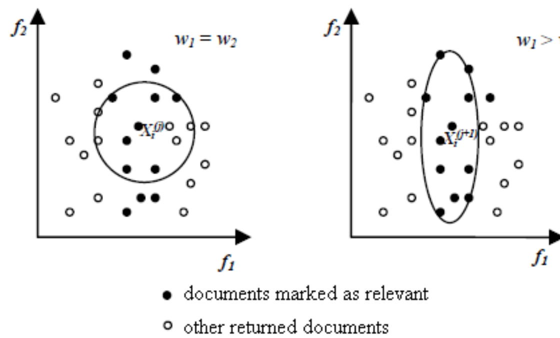

Figure 3: Schematic ilustration of the RFE algorithm, as showed in [17]

C. Robertson-Sparck-Jones algorithm

In the Robertson-Sparck-Jones model of information retrieval [10], the terms in a corpus are all assigned relevance weights, which are updated for any particular query. For positive feedback, the relevance weights will be very small (and the distance between the query video and the target video will be 0); for negative feedback, the relevance weights will be significant. Initially, all the weights are equal to 1, later being updated according to the users feedback. After user's feedback the distance between two videos will become:

$$Dist(X,Y) = W_i \sqrt{\sum_{i=1}^{d} (X_i - Y_i)^2} \tag{5}$$

D. Support Vector Machines

Support vector machines [14] have become extremely successful in domains as pattern classification or regression. These represent neural networks with two layer architecture that constructs a hyperplane or set of hyperplanes in a high dimensional space, which can be used for classification tasks. Support Vector Machine (SVM) models are close to classical multilayer perceptron neural networks with two layers, but the hidden layer uses kernel functions, that transform low level dimension into high level dimension to simplify the problem [8]. The kernel function can be an inner product, Gaussian basis function, polynomial, or any other function that agrees Mercer's condition.

The central idea of SVM is to adjust a discriminating function so that it makes optimal use of the reparability information of boundary cases. Given a set of cases which belong to one of two classes, training a linear SVM consists in searching for the hyperplane that leaves the largest number of cases of the same class on the same side, while maximizing the distance of both classes from the hyperplane.

Implementations of SVM method in Relevance Feedback are proposed in [15] and [16].

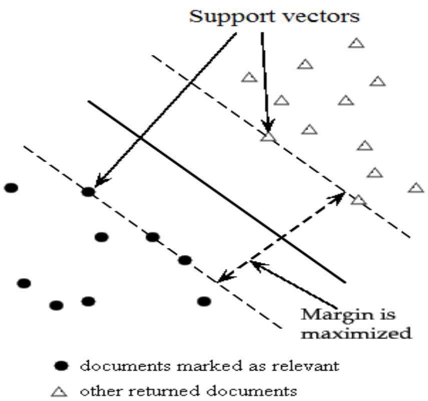

Figure 4. Maximum-margin hyperplane and margins for an SVM trained with samples from two classes.

In our experiments we use linear kernel function:

$$o_k = K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \quad (6)$$

where x is the feature vector and K() is the kernel function

V. HIERARCHICAL CLUSTERING RELEVANCE FEEDBACK

We propose a relevance feedback approach which is based on hierarchical clustering [18] (HCRF). HCRF represents a classical method of data analysis, which aims to partition the observations into clusters. The number of clusters varies form one iteration to another, driven by the merging (agglomerative or bottom-up clustering) or by the division (division or top-down clustering) of some of the existing clusters. The hierarchical agglomerative clustering (HAC) successively searches for the most similar clusters in the current partition; these clusters are merged and thus the total number of clusters in the partition decreases by one. There are several classical ways for measuring the cluster similarity (mean distance, minimal variance, etc..), each of them relying on some assumptions about the nature of the observation set.

The HAC can produce a hierarchical ordering (called dendrogram) of the observation/ clusters, which may be informative for data display and discovery of data relations. Small clusters (helpful for data structure discovery) can be generated and can survive across the iterations if they are different enough. There is no assumption related to the shape of the clusters.

The fundamental iteration of the HCRF can be described as follows:

- initialize the clusters with the video initially labeled by the user in the current relevance feedback iteration (each cluster consists of a single video);
- perform the hierarchical aggregative clustering based on the centroid distance, by merging the most similar clusters within each category (relevant/ non-relevant); the clustering is stopped when the number of remaining clusters becomes relevant for the video categories present within the retrieved image set (a heuristic choice is to set the minimal number of clusters equal to a quarter of the number of video within a retrieved batch, or viewing screen);
- for each new browsing batch of retrieved videos, do classify the next videos as relevant or non-relevant with respect to the existing clusters by the same hierarchical aggregative clustering approach presented before;
- if needed, acquire new relevance feedback information from the user and repeat the previous step.

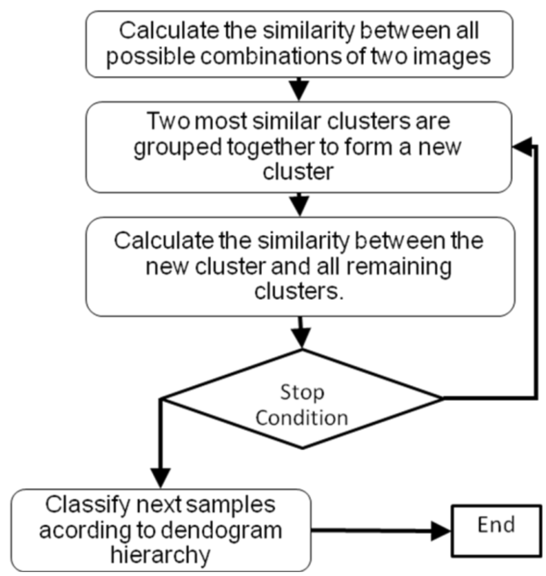

Figure 5. Schematic ilustration of the Hierarchical Clustering algorithm.

The proposed hierarchical clustering relevance feedback (HCRF) starts from the basic idea that the video content descriptor is good enough, such that within the first retrieved videos are at least some relevant videos that can be used as positive feedback by the user. Also, in most cases, there is at least one non-relevant video that can be used as negative feedback.

The retrieved videos are presented to the user in batches (corresponding to the videos that are simultaneously shown on the screen) and the user browses through the query results by viewing successive batches. Instead of modifying the query or the similarity metric, as most RF algorithms do, we propose to simply cluster (group, classify) the remaining retrieved videos with respect to the user-labeled videos. At each feedback iteration, the retrieved videos that are in the next browsing batch will be clustered by a hierarchical agglomerative clustering algorithm.

VI. Experimental results

We tested the effectiveness of the proposed relevance feedback algorithms on a database consisted in 91 hours of video, containing 20h30m of animated movies (long, short clips and series), 15m of TV commercials, 22h of documentaries (wildlife, ocean, cities and history), 21h57m of movies (long, episodes and sitcom), 2h30m of music (pop, rock and dance video clips), 22h of news broadcast and 1h55min of sports (mainly soccer) (a total of 210 sequences, 30 per genre).

The visual video content description for the color videos is implemented using three types of features: color, action and contour based. We test several combinations of features, thus: color – action, contour feature alone and color, action and contour all together.

The user feedback is automatically simulated from the known class member-ship of each video (in this scenario video footage is labeled according to video genre). According to this known class membership, the simulation of relevance feedback provides the user response accordingly. This approach allows a fast and extensive simulation (which could not be realized otherwise) but lacks the inherent errors, change of mind and unexpected connections that a real user could be subject to.

TABLE I. Mean precision improvement with relevance feedback

Initial Descriptor	40.82%
Rocchio	58.20%
Robertson/Starck-Jones	55.83%
FRE	68.48%
Support Vector Machines	70.28%
Hierarchical Clustering RF	**76.61%**

Precision represents the percent of correctly retrieved videos within the total number of retrieved videos. Overall, the medium precision improvement, computed on all categories, is presented in Table I (computed using the before defined descriptors).

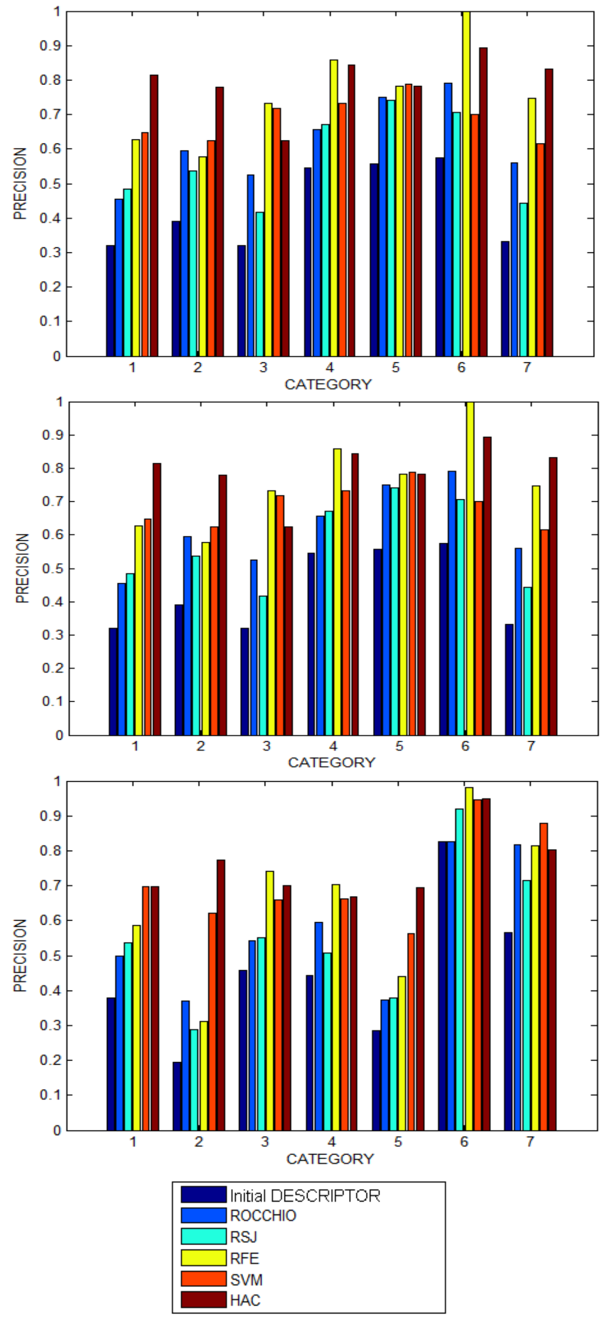

Figure 5: Precision values per genre category curves for different descriptors (from top to bottom): Color & Action, Contour, Contour & Color & Action; showing the behavior of RF methods after one iteration. On all plots we have the original method (bluemarin bar), Rochio (blue bar) Robertson Spark Jones RF (cyan line), FRE RF (yellow bar), SVM (red bar) and HCRF (magenta bar). The categories are: 1 – Animated, 2 – Advertising videos, 3 – Documentaries, 4 – Movies, 5 – Video Clips, 6 – News, 7 – Sport Videos.

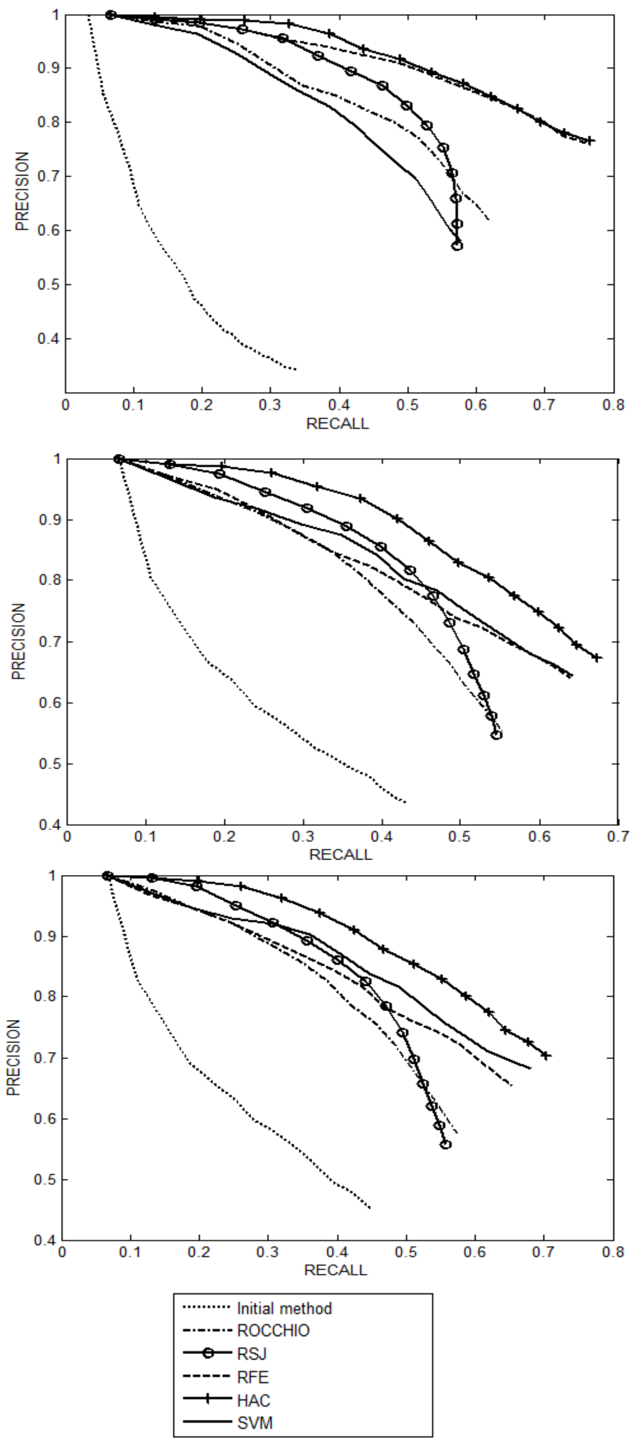

Figure 6: Precision-recall curves per genre category curves for different descriptors (from top to bottom): Color & Action, Contour, Color & Action & Contour showing the behavior of RF methods after one iteration. On all plots we have the original CHD (dotted line), Robertson Spark Jones RF (dash-dotted line), FRE RF (dashed line), Rocchio RF (continuous line with circle marks) , SVM RF (continous line) and HC RF (upper continuous line).

We computed the precision improvement per movie category. The results are summarized in Figure 5.

Charts show that SVM, hierarchical clustering and RFE improve the system performance with the highest percentage: hierarchical clustering has the maximum percentage for nine experiments (for animated, advertising, video clips and sports), RFE for eight experiments (news, movies and documentary categories) and SVM for four experiments (animated and sports).

The largest increase in system performance is obtained on news videos using hierarchical clustering: from 17,7% to 82%, while the lowest increase is obtained for movies and documentaries (from 32 to 42 percent and from 54 to 82 percent). The reason why the news category achieves a high performance is that the class is very compact, while movies and documentaries are more diversified.

Since the correct class membership is known for any video within the database, we evaluate the quantitative, objective retrieval performance of the proposed methods via the classical precision-recall curves [11], [12].

In this case, the precision is the percent of correctly retrieved videos within the total number of retrieved videos while the recall is the percent of correctly retrieved videos with respect to the total number of relevant videos within the database.

The precision-recall curve plots the precision for all the recall rates that can be obtained according to the current video class population; the evaluation process is repeated considering each video from the database as query video and retrieving the remainder of the database accordingly. Figure 6 presents the performance of the proposed RF algorithm, compared with classical RF methods for the tested video database.

All the presented results clearly show that the HCRF approach performs better than classical relevance feedback methods. For 7 showed videos the largest increase in performance is obtained in the second experiment (from 57% to 90%, while SVM and FRE achieved 82% and 84%).

We obtained a good performance for the first experiment too, from 58% to 93,6% but FRE has a similar increase in performance (92.5%). The performance obtained from classical algorithms (Rocchio and Robertson-Sparck-Jones) is lower with 10 to 30 percents than the hierarchical clustering approach.

The experiments showed that hierarchical clustering approach works well for a large range of simultaneously shown video (size of the video batch), between 10 and 25. For the presented experiments this parameter was set to 15.

VII. CONCLUSIONS

We improved the performance of classic video retrieval systems by employing several relevance feedback schemes: Rocchio RF, Robertson Spark Jones, Feature Relevance Estimation RF, SVM and hierarchical clustering RF on a video database.

The proposed hierarchical clustering relevance feedback (HCRF) outperforms classical RF algorithms (such as Rocchio, RFE or Robertson Sparck Jones) in terms of speed, storage requirement and accuracy for video databases. For some movies categories we obtained improvements from 20 to more than 80 percent.

SVM has a similar performance for many categories of movies, but different computational complexity. It works well when the dimension of the feature array is higher (more than 300 items). Feature Relevance Estimation has a high improvement for many categories like documentaries, movies and news, but low increase of performance for other types of videos. Future work consists in performing experiments using other variants of hierarchical clustering algorithms. We are trying to improve classification for more sophisticated video categories such as movies and documentaries.

ACKNOWLEDGMENT

Part of this work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

REFERENCES

[1] Singh, R., Seltzer, M.L., Raj, B., and Stern, R.M. "Speech in Noisy Environments: Robust Automatic Segmentation, Feature Extraction, and Hypothesis Combination," IEEE Conference on Acoustics, Speech and Signal Processing, Salt Lake City, UT, May, 2001.

[2] Sato, T., Kanade, T., Hughes, E., and Smith, M. Video OCR for Digital News Archive. In Proc. Workshop on Content-Based Access of Image and Video Databases. (Los Alamitos, CA, Jan 1998), 52-60.

[3] S. Sav, H. Lee, A. Smeaton, N. O'Connor, and N. Murphy. Using Video Objects and Relevance Feedback in Video Retrieval. In Proceedings of SPIE (SPIE, Bellingham, Wa), 2005.

[4] S. Aksoy, O. Cavus. A relevance feedback technique for multimodal retrieval of news videos. In Proceedings of EUROCON, Belgrade, Serbia & Montenegro, November 21-24, 2005.

[5] Cees G.M. Snoek and Marcel Worring "Multimodal Video Indexing: A Review of the State-of-the-art", ISIS technical report series, Vol. 2001-20, December 2005

[6] Yong Rui, T. S. Huang, M. Ortega, M. Mehrotra, S. Beckman, "Relevance feedback: a power tool for interactive content-based image retrieval", IEEE Trans. on Circuits and Video Technology, vol. 8, no. 5, Sept. 1998, pp. 644-655, 1998.

[7] B. Ionescu, D. Coquin, P. Lambert, V. Buzuloiu: "A Fuzzy Color-Based Approach for Understanding Animated Movies Content in the Indexing Task," Eurasip Journal on Image and Video Processing, doi:10.1155/2008/849625, 2008.

[8] C. Rasche: "An Approach to the Parameterization of Structure for Fast Categorization," Int. Journal of Computer Vision, 87(3), pp. 337-356, 2010.

[9] J. Rocchio: "Relevance Feedback in Information Retrieval", in The Smart Retrieval System – Experiments in Automatic Document Processing, G. Salton (Ed.), Prentice Hall, Englewood Cliffs NJ, 1971, pp. 313-323.

[10] S. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, M. Gatford: "Okapi at TREC-3". in Overview of the Third Text Retrieval Conference (TREC-3), D.K. Harman (Ed.), Gaithersburg, MD: NIST, 1995 (NIST Special Publication 500-225), pp. 109-126, 1996.

[11] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, A. Yamada: "Color and Texture Descriptors", IEEE Trans. on Circuits and Systems for Video Technology, vol. 11, no. 6, Jun. 2001, pp. 703-715, 2001.

[12] J. M. Martinez: "Standards - MPEG-7 Overview of MPEG-7 description tools, part 2". IEEE MultiMedia, vol. 9, no. 3, pp. 83–93, 2002.

[13] N. Maillot: "Ontology Based Object Learning and Recognition", PhD thesis, Université de Nice Sophia Antipolis, 2005.

[14] V.N. Vapnik, Statistical Learning Theory. New York: John Wiley & Sons, 1998.

[15] Rujie Liu, YuehongWang, Takayuki Baba, Daiki Masumoto, Shigemi Nagata 2008 SVM-based active feedback in image retrieval using clustering and unlabeled data, 41 (2008) 2645 – 2655

[16] Shuang Liang, Zhengxing Sun "Sketch retrieval and relevance feedback with biased SVM classification", Pattern Recognition Letters 29 (2008) 1733–1741

[17] S. Santini, Exploratory Image Databases Content-based Retrieval, Academic Press, 2001.

[18] W. J. Krzanowski, Principles of Multivariate Analysis: A User's Perspective, Clarendon Press, Oxford, 1993.