Fisher Kernel based Relevance Feedback for Multimodal Video Retrieval

Ionuţ Mironică
LAPI, University Politehnica of Bucharest, Romania.
imironica@imag.pub.ro

Bogdan Ionescu
LAPI, University Politehnica of Bucharest, Romania.
bionescu@imag.pub.ro

Jasper Uijlings
DISI, University of Trento, Italy.
jrr@disi.unitn.it

Nicu Sebe
DISI, University of Trento, Italy.
sebe@disi.unitn.it

ABSTRACT

This paper proposes a novel approach to relevance feedback based on the Fisher Kernel representation in the context of multimodal video retrieval. The Fisher Kernel representation describes a set of features as the derivative with respect to the log-likelihood of the generative probability distribution that models the feature distribution. In the context of relevance feedback, instead of learning the generative probability distribution over all features of the data, we learn it only over the top retrieved results. Hence during relevance feedback we create a new Fisher Kernel representation based on the most relevant examples. In addition, we propose to use the Fisher Kernel to capture temporal information by cutting up a video in smaller segments, extract a feature vector from each segment, and represent the resulting feature set using the Fisher Kernel representation. We evaluate our method on the MediaEval 2012 Video Genre Tagging Task, a large dataset, which contains 26 categories in 15.000 videos totalling up to 2.000 hours of footage. Results show that our method significantly improves results over existing state-of-the-art relevance feedback techniques. Furthermore, we show significant improvements by using the Fisher Kernel to capture temporal information, and we demonstrate that Fisher kernels are well suited for this task.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: [Indexing methods]; H.3.3 [**Information Search and Retrieval**]: [Retrieval models, query formulation, relevance feedback]

General Terms

Algorithms, Performance, Experimentation

Keywords

Relevance feedback, Fisher kernels, multimodal video retrieval

1. INTRODUCTION

The actual challenge of the existing information retrieval systems is in their capability of identifying and selecting only relevant information, according to some user specifications. This issue became more critical due to increase of technology, e.g. portable multimedia terminals, wireless transmission protocols, imaging devices which basically makes the information accessibile from everywhere. In order to improve performance, existing systems are turning towards multimodal approaches attempting to exploit the benefits of fusing various modalities such as text, visual and audio. Despite the high variability of automatical content descriptors used and of the classification techniques, Content Based Video Retrieval Systems (CBVR) are inherently limited by the gap between the real world and its representation through computer vision techniques [1].

An effective way to narrow the semantic gap is to use the user's feedback in the retrieval process, which is known as Relevance Feedback (RF). A general RF scenario can be formulated as follow: for a certain retrieval query, the user marks what results are relevant and non-relevant. Then, the system automatically computes a better representation of the information and/or retrains the classifier to better refine results. Relevance feedback can go through one or more iterations of this sort. This basically improves the system's response based on query related ground-truth.

In this paper, we propose a new RF approach for video genre retrieval, using a combination of Fisher Kernels with SVM Classifiers. Fisher Kernels (FK) are a powerful framework which combines the advantages of a generative algorithm with the strengths of discriminative approaches [2]. The main idea of FK is to describe a signal with a gradient vector derived from a generative probability model (Gaussian Mixture Model - GMM) and then to train this representation with a discriminative classifier (in most of the cases SVM). The Fisher Kernels have been successfully applied to many fields from image categorization [3], to audio indexing [5] and handwritten word-spotting [4], but, to our knowledge, the FK have never been used in Relevance Feedback, or in video classification.

In order to describe a document, most of the RF strategies use a single feature vector. However, video documents can be considered as a sequence of scenes, and the features from each scene can be used to model and retrieve the video content. Because we use the Fisher Kernel framework, we can retain some form of temporal relations of the video scenes in our relevance feedback approach.

Experimental tests conducted on the large video database MediaEval Genre Tagging task 2012 [6] and using current state-of-the-art multimodal video descriptors, prove that the proposed RF increases retrieval performance and outperforms other classic approaches. In addition, the proposed approach allows a fast implementation similar to a classical SVM RF strategy, but with a higher increase of performance. We also propose several modifications to the original framework that can boost the accuracy of the RF algorithm.

2. RELATED WORK

The idea of relevance feedback is to take advantage of the user's input on the initially returned results for a given query and to use this information to refine and improve the quality of the results. Relevance feedback has proven to increase retrieval accuracy, and to give more personalized results for the user [10] [11] [12] [13] [15]. Recently, a relevance feedback track was organized by TREC to evaluate and compare different relevance feedback algorithms for text descriptors [7]. However, relevance feedback was successfully used not only for text retrieval, but also for image features [11] [12] [13] [15] and multimodal video features [10] [21].

Most of the relevance feedback algorithms can be divided in two main classes: those that change the feature's representation, and, those that use a re-learning strategy with a classifier.

One of the earliest and most successful RF algorithms is the Rocchio algorithm [9] [10]. Using the set of R relevant and N non-relevant documents selected from the current user relevance feedback window, the Rocchio algorithm modifies the feature of the initial query by adding the features of positive examples and subtracting the features of negative examples to the original feature.

The Relevance Feature Estimation (RFE) algorithm [11] assumes that for a given query, according to the user's subjective judgment, some specific features may be more important than other features. The idea of the re-weighting strategy is to analyze the relevant objects in order to understand which dimensions are more important than others. Every feature has an importance weight computed as $w_i = 1/\sigma$ where σ denotes the variance of relevant documents. Therefore, features with higher variance with respect to the relevant queries become less important than elements with a reduced variation.

More recently, machine learning techniques found their application with relevance feedback approaches. In these approaches, the relevance feedback problem can be formulated as a two class classification of the negative and positive samples. After a training step, all the documents are ranked according to the classifiers's confidence level. Some of the most successful techniques use Support Vector Machines [12], Nearest Neighbor [13], classification trees, e.g. Random Forest [15], or boosting techniques, e.g. AdaBoost [14].

However, all these techniques have problems when there is only a limited number or an asymmetric number of positive and negative feedback samples provided by the user. There have been several attempts to overcome this. More recent approaches to relevance feedback include Biased Discriminant Euclidean Embedding [17] and Active Reranking for Web Image Search [18]. However, all of these approaces have only been applied to image datasets.

In video retrieval, most of relevance feedback approaches have focused on pseudo-relevance feedback. In general, pseudo-relevance feedback algorithms assume that a substantial number of video shots in the top of the ranking are relevant [19]. The information associated with these top-ranked pseudo-relevant shots is then used to update the initial retrieval results.

In this paper, we propose a new method, denoted Fisher Kernels Relevance Feedback (FKRF). We will show that we can improve the performance of classical RF retrieval systems by using the Fisher kernel method. Furthermore, the use of the Fisher kernel representation enables us to represent a complete video while retaining a form of temporal information.

3. PROPOSED FISHER KERNEL RELEVANCE FEEDBACK

3.1 Fisher Kernel Theory

The Fisher kernel were designed as a framework to combine the benefits of generative and discriminative approaches. The general idea is to represent a signal as the gradient of the probability density function that is a learned generative model of that signal. Intuitively, such Fisher vector representation measures how to modify the parameters of the probability density function in order to best fit the signal, similar to the measurements in a gradient descent algorithm for fitting a generative model. The Fisher vector is subsequently used in a discriminative classifier. In this paper we follow [2] and use a Gaussian Mixture Model (GMM) followed by a linear SVM.

Let $X = \{x_1, x_2, ..., x_T\}$ be a set of T multimodal video descriptors. Now X can be represented by its gradient vector with respect to a Gaussian Mixture Model u_λ with parameters λ:

$$G(X)_\lambda = \frac{1}{T} \bigtriangledown_\lambda \, log\,(u_\lambda(X)) \qquad (1)$$

The gradient vector is, by definition, the concatenation of the partial derivatives with respect to the model parameters. Let μ_i and σ_i be the mean and the standard deviation of i's Gaussian centroid, $\gamma(i)$ be the soft assignment of descriptor x_t to Gaussian i, and let D denote the dimensionality of the descriptors x_t. $G^x_{\mu,i}$ is the D-dimensional gradient with respect to the mean μ_i and standard deviation σ_i of Gaussian i. Mathematical derivations lead to [3]:

$$G^x_{\mu,i} = \frac{1}{T\sqrt{\omega_i}} \sum_{t=1}^{T} \gamma(i) \frac{x_t - \mu_i}{\sigma_i} \qquad (2)$$

$$G^x_{\sigma,i} = \frac{1}{T\sqrt{2\omega_i}} \sum_{t=1}^{T} \gamma(i) \left[\frac{(x_t - \mu_i)^2}{{\sigma_i}^2} - 1 \right] \qquad (3)$$

where the division between vectors is a term-by-term operation. The final gradient vector G^x is the concatenation of the $G^x_{\mu,i}$ and $G^x_{\sigma,i}$ vectors, for $i = 1...K$.

3.2 Proposed Fisher Kernel RF Algorithm

Our relevance feedback method works as follows. Using a single video as query, we rank all videos using a nearest-neighbor strategy. Then, the user selects from the top n videos which ones are relevant or which ones are not, where n is typically small (20 in our experiments). We learn a generative Gaussian Mixture model from the first n retrieved documents. Then we re-represent the top k videos using a Fisher Kernel representation with respect to this GMM, where k is typically large (2000 in our experiments). We only consider the top k as it is unlikely that relevant videos are ranked lower in the initial ranking. Afterwards, we train an SVM on the Fisher vectors of the top n user labeled results. We apply this SVM on the top k videos to obtain a final ranking.

The algorithm is given in Algorithm 1. We now briefly describe the details for re-representing the features after relevance feedback using the Fisher Kernel and the subsequent learning procedure.

Altering features after user's feedback. After the initial query using nearest-neighbor search, we train a Gaussian Mixture model on the features of the top n videos, regardless of their true relevance. In a practical application this allows the training of the Gaussian Mixture model in the background during the time that the user is giving feedback. For optimization reasons we initialize the centroids with a kmeans output. An important choice for the Fisher Kernel representation is the number of clusters c. As for each cluster the dimensionality of the representation doubles, for a practical system the number of clusters has to be low. We experiment with a value of c between 1 and 5 in Section 5.2.

The size of the Fisher Kernel representation is twice the size of the original feature times the number of clusters c. To make the Fisher Kernel computationally feasible, we first apply PCA on the original feature vectors of the documents. We compute PCA individually on each feature type and reduce the dimensions by 10%. After having obtained the mixture model, we convert the original features of the top k videos into the Fisher Kernel representation using Equations 2 and 3. For both the GMM clustering and the Fisher projection we use the software obtained from [3].

Finally, we perform normalization on the Fisher vectors as [2] has found this to significantly increase performance. In our method we experiment with the following normalization strategies: L1 and L2 normalizations, power normalization ($f(x) = sign(x)\sqrt{\alpha|x|}$), logarithmic normalization ($f(x) = sign(x)log(1 + \alpha|x|)$) and combinations of them.

Training - reranking step. We use the Fisher representations of the top n videos, along with the labels obtained using feedback from the user, to train a two-class SVM classifier. SVMs are appropriate for relevance feedback as they are relatively robust to the situation in which only few training examples are available. Indeed, SVMs have been successfully used in several RF approaches [12]. In our experiments, we test two types of SVM kernels: a fast linear kernel and the RBF nonlinear kernel. While linear SVMs are very fast in both training and testing, SVMs with an RBF kernel are more accurate in many classification tasks.

Algorithm 1: The Fisher Kernels Relevance Feedback Algorithm

```
Initial parameters:
  Labeled Sample set: $X_i$ and labels $Y_i$;
  Unlabeled Sample set: $X_r$;
  SVM Classifier parameters ($C$, $\gamma$);
  $n$: the window size;
Start:
  do 10% PCA reduction for all multimodal features;
Altering features step:
  Compute GMM centroids for $X_i$;
  for $x \subset Xi$ do
      compute $FK(x) = FK(x, GMM)$;
      normalize $FK(x)$;
Training - reranking step:
  train SVM($C$, $\gamma$) using $FK$ features;

  for $x \subset Xr$ do
      compute $FK(x) = FK(x, GMM)$;
      normalize $FK(x)$;
      compute $h(x)$ = SvmConfidenceLevel($FK(x)$);
  sort $h(x)$ values;
  show new ranked list according to $h(x)$ values;
```

3.3 Frame Aggregation with Fisher kernel

Most of content based systems involve two main steps: feature extraction and document ranking. The first step mainly consists of computing one feature per document that needs to capture as many relevant characteristic for that document category as possible. For video documents, most of the approaches compute a feature for each frame, and then aggregate all the features in one descriptor by computing the mean, dispersion or other statistics over all the frames. But, by aggregating these statistics in these ways, the notion of time is lost. Alternatively, we can represent a video by multiple vectors and to compute the distance between two sets of points using, for example, the Earth Mover distance [24]. However, using such a metric involves a huge computational cost for large databases.

By using the Fisher kernel representation, we obtain a natural solution to the problem above. The Fisher Kernel was originally designed to map multiple vectors into a fixed length representation, and this approach is exactly what we need for this problem. It takes advantage of the expressive power of generative models to map sequences of features of variable length, such as video sequences, into a fixed length representation.

For cutting up the video into temporal fragments, one approach is to divide the video document into frames and to compute a visual descriptor for each video frame. However, for large multimedia databases the number of frames is huge (25 frames per second and thousands of hours of video footage) and this approach can create computational problems. In order to efficiently browse through the significant video content, summarization is required [27]. So, we extract a small collection of salient images (keyframes) that best represent the underlying content [27].

We train the Gaussian Mixture model on the features of the top n videos. Once the generative model is trained, for every training sequence of feature vectors $X_i = \{x_1; x_2; ..; x_T\}$, composed of T feature vectors we transform it into a vector of fixed dimension. The only difference between the previous

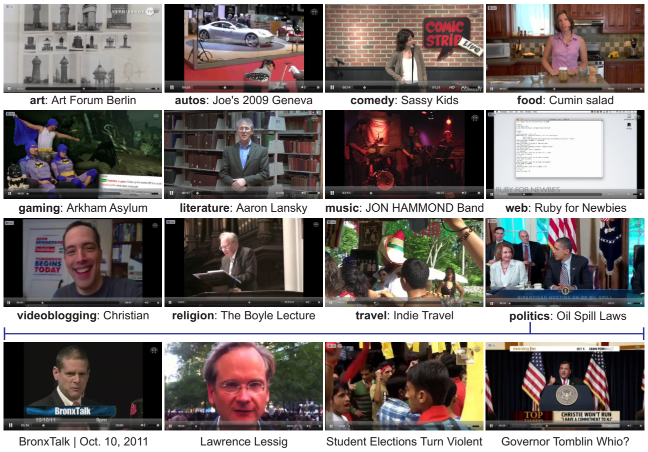

Figure 1: Image examples for several video genres. The bottom images correspond to videos from the same genre category, i.e. ”politics” (source blip.tv).

approach and this one regards on what data the GMM has learned. Instead of using one global aggregated video feature, we will use more features per document. The resulting Fisher kernel representation will have the same number of dimensions. Experiments from Section 5.5 will show the performance of temporal Fisher Kernels on the Relevance feedback problem.

4. EXPERIMENTAL SETUP

4.1 Dataset

The validation of the proposed content descriptors is carried out in the context of the MediaEval 2012 Benchmarking Initiative for Multimedia Evaluation, the Video Genre Tagging Task [6]. This task addresses the automatic categorization of web media genres used with the blip.tv platform (see http://blip.tv/). For testing, we use the MediaEval 2012 Video Genre Tagging dataset consisting of 15000 sequences (up to 2000 hours of video footage), labeled according to 26 video genre categories, namely: art, autos and vehicles, business, citizen journalism, comedy, conferences and other events, documentary, educational, food and drink, gaming, health, literature, movies and television, music and entertainment, personal of auto-biographical, politics, religion,school and education, sports, technology, environment, mainstream media, travel, videoblogging, web development and ”default” category (accounts for movies which cannot be assigned to neither one of the previous categories).

The main challenge of this task is the diversity of videos that contain high level concepts for videos genres, each genre category has a high variety of video materials. Figure 1 illustrates image examples from the dataset.

4.2 Evaluation

In our experiments we consider the scenario that user feedback is automatically simulated from the known class membership of each video document (ground truth is provided with the databases). This approach allows a fast and extensive simulation which is necessary to evaluate different methods and parameter settings. Such simulations are common practice for RF [11] [12] [14].

To assess the retrieval performance, we use several measures. First, we compute the classical precision and recall. Precision is the fraction of retrieved documents that are relevant to the search (measure of false positives) and recall is the fraction of the documents that are relevant to the query and successfully retrieved (measure of false negatives). The system retrieval response is assessed with the precision-recall curves, which plots the precision for all the recall rates that can be obtained according to the current document class population. Second, to provide a global measure of performance we determine the overall Mean Average Precision - MAP as the area under the uninterpolated precision-recall curve.

In our evaluation we systematically consider each document from the database as a query document and retrieve the remainder of the database accordingly. Precision, recall and MAP are averaged over all retrieval experiments. Experiments were conducted for various browsing top n, ranging from 10 to 30 documents. For space and brevity reasons, in the following we shall present only the results using the top 20 videos per window. The general observations in this paper hold for all values of n.

4.3 Content descriptors

For video descriptors we have used a broad range of descriptors including: visual, audio and text. Competitive results have been obtained using these descriptors on MediaEval Genre Tagging task 2012 [6].

Audio features

Block-based audio features (11,242 values) [28] capture the temporal properties of the audio signal. We choose a set of audio descriptors that are computed from overlapping audio blocks. On each block, we compute the Spectral Pattern which characterizes the soundtrack's timbre, delta Spectral Pattern which captures the strength of onsets, variance delta Spectral Pattern which represents the variation of the onset strength over time, Logarithmic Fluctuation Pattern which captures the rhythmic aspects, Spectral Contrast Pattern, Correlation Pattern which compute the temporal relation of loudness changes and timbral features: Local Single Gaussian Model and Mel-Frequency Cepstral Coefficients. Sequence aggregation is achieved by taking the mean, variance and median over all blocks.

Standard audio features (196 values) [29] [30] - we used a set of general-purpose audio descriptors: Linear Predictive Coefficients (LPCs), Line Spectral Pairs (LSPs), MFCCs, Zero-Crossing Rate (ZCR), spectral centroid, flux, rolloff and kurtosis, augmented with the variance of each feature over a certain window (we used the common setup for capturing enough local context that is equal to 1.28s). For a clip, we take the mean and standard deviation over all frames.

Visual descriptors

MPEG-7 related descriptors(1,009 values) [31] - we adopted standard color and texture-based descriptors such as: Local Binary Pattern (LBP), autocorrelogram, Color Coherence Vector (CCV), Color Layout Pattern (CLD), Edge Histogram (EHD), Scalable Color Descriptor (SCD), classic color histogram (hist) and color moments. For each sequence, we aggregate the features by taking the mean, dispersion, skewness, kurtosis, median and root mean square statistics over

Table 1: Comparison between feature accuracy (MAP) using different metrics without RF.

Feature	Manhatan	Euclidian	Mahalanobis	Cosinus	Bray Curtis	Chi Square	Canberra
`Hog Features`	17.02%	**17.18**%	17.07%	17.00%	17.10%	17.07%	16.67%
`Structural Features`	10.87%	10.55%	11.14%	2.18%	10.92%	11.58%	**14.82**%
`MPEG 7 related`	12.37%	10.85%	21.14%	08.69%	13.34%	13.34%	**25.97**%
`Standard Audio Features`	7.76%	7.78%	**29.26**%	15.28%	7.78%	8.04%	1.58%
`Block-based audio`	19.33%	19.58%	20.21%	**21.23**%	19.71%	19.99%	20.37%
`Text Features`	8.32%	7.15%	5.39%	17.64%	**20.40**%	9.83%	9.68%

all frames.

Global HoG (81 values) [32] - from this category, we compute global Histogram of oriented Gradients (HoG) over all frames.

Structural descriptors (1,430 values) - the structural description [33] is based on a characterization of geometric attributes for each individual contour, e.g. degree of curvature, angularity, circularity, symmetry and "wiggliness", as proposed in [33]. These descriptors were reported to be successfully employed in tasks such as the annotation of photos and object categorization [34].

In this work, we decided not to use Bag of Words strategies. In preliminary experiments we found that in order to get results as good or better than the other visual features, we need large dictionaries that create computational problems for the large dataset we use.

Text descriptors

TF-IDF (extracted with automatical speech recognition algorithms ASR, with 3,466 values, provided by the MediaEval organizers [35]) - we use the standard Term Frequency-Inverse Document Frequency approach. First, we filter the input text by removing the terms with a document frequency less than 5%-percentile of the frequency distribution. We reduce further the term space by keeping only those terms that discriminate best between genres according to the 2-test. We generate a global list by retaining for each genre class, the m terms (e.g. m = 150 for ASR) with the highest 2 values that occur more frequently than in complement classes. This results in a vector representation for each document that is subsequently cosine normalized to remove the influence of the length of transcripts.

We used in our framework eight combinations of multimodal video descriptors: Visual (1 - MPEG-7 related descriptors, 2 - Hog Features, 3 - Structural descriptors, 4 - Combination of All Visual descriptors), Audio (5 - Standard audio features, 6 - Block-based audio descriptor), Text descriptors (7 -TF-IDF - ASR-based), and 8 - combination of all of them. All the visual and audio descriptors are normalized to L_∞ norm, and text descriptors to cosine normalization.

5. RESULTS

In the following subsections, we present our experiments. The first experiment (Section 5.1) motivates the choice of the best metric that provides the best accuracy for each feature. In the second experiment (Section 5.2), we study the influence of Fisher Kernel parameters on system's accuracy, and in Section 5.3 we compare our work with state-of-the-art techniques. In Section 5.4 we compare our method with a Fisher Kernel representation by learning a GMM on *all* the data and in Section 5.5 we illustrate the advantage of Fisher Kernels approach on video RF using more than one descriptor per video document.

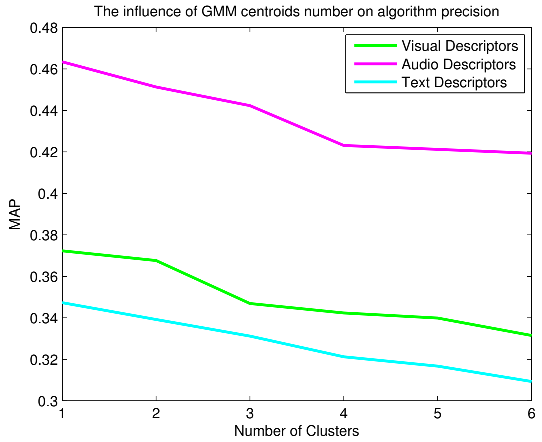

Figure 2: The influence of GMM centroids number on system performance for T=1.

5.1 Evaluating feature metrics

Some distance measures are better adapted to the structure of the descriptor than others [22]. In this work we have tested the performance of several metrics. We made the assumption that a better initial performance will generate a better relevance feedback algorithm performance [22].

We tested a broad variety of metrics [23]: Euclidean (L2), Manhattan (L1) (particular cases of the Minkovski distance), probabilistic divergence measures: Canberra [26]; intersection family: Cosine Distance, Chi-Square distance used in machine learning and data clustering and Mahalanobis [25]. The performance of metrics is presented in Table 1. These experiments were performed on the entire MediaEval 2012 dataset.

We conclude that each feature has its own preferred metric. In the rest of our experiments we use for each feature its best metric as indicated in Table 1 (bolded results).

5.2 RF using Fisher Kernels

In this experiment we study the influence of Fisher Kernel parameters on the system's performance.

We first analyze the influence of the number of Gaussian centroids. Figure 2 presents the variation of MAP using a different number of Gaussian centroids. It can be observed that the best results are obtained using only a single Gaussian centroid. In this case the size of Fisher Kernel descriptors will be 2 times bigger than the document descriptor.

Secondly, we presents the influence of Fisher normaliza-

Table 2: The influence of different normalization algorithms on system's performance (mean average precision values).

Features Normalization	Visual	Audio	Text
Without normalization	37.25%	38.68%	31.13%
L1	36.82%	37.97%	29.83%
L2	39.22%	41.94%	30.51%
Log Norm	38.61%	42.01%	**35.07**%
PN	38.51%	41.37%	34.93%
PN + L2 Norm	39.20%	42.98%	30.12%
PN + L1 Norm	**39.46**%	**43.23**%	31.71%

tion strategies on system performance. In [3], it was demonstrated that some normalization strategies can improve the performance of Fisher Kernels. The results are presented in Table 2.

It can be observed that using the combination of L1 normalization - alpha normalization we obtained the best results for visual and audio features, while the highest performance for text features is obtained with logarithmic normalization. Another observation is that using the L1 normalization alone, the results are lower than in the case when L1 is used in combination with other normalizations.

In order to compare our algorithm with other relevance feedback approaches, we have selected the settings that provide the greatest improvement in performance: one GMM centroid, L1 normalization with alpha normalization for audio and visual descriptors and logarithmic normalization for text descriptors. We also used 2 SVM kernels: a linear SVM classifier and a nonlinear RBF kernel.

5.3 Comparison to state-of-the-art techniques

In the following, we compare our approach against other validated algorithms from the literature, namely: the Rocchio algorithm [9], Relevance Feature Estimation (RFE) [11], Support Vector Machines (SVM) [12], AdaBoost (BOOST) [14], Random Forests (RF) [15] and Nearest Neighbor [13].

Figure 3 presents the precision-recall curves after relevance feedback for different descriptor categories. Generally, all RF strategies provide significant improvement in retrieval performance compared to the retrieval without RF (see the dashed black and blue lines in Figure 3). Better performance is obtained with audio descriptors, while text and visual descriptors have similar performance.

The highest performance is obtained using standard audio descriptors, with an increase of MAP from 29.35% (without RF) to 46.34% and with all combined features from 30.29% to 46.80%.

We present in Table 3 the MAP values for different features combinations. The FKRF approach has the highest values for most of the cases, except for the combination of all visual descriptors, where the random trees RF achieve the highest performance values. The highest increase in system performance is obtained using MPEG 7 descriptors, increase of 4 MAP percents (from 40.80% using FKRF RBF to 36.85% with random forests) and block based audio (from 43.96% using FKRF Linear to 39.87% using Boost RF). At the other end, the smallest increase in performance is obtained for text features (from 45.80% using FKRF RBF to 45.31 using random forests).

In most of the cases, RFE and random forests provide good results, but our approach is better. We conclude that the proposed approach improves the retrieval performance, outperforming some other existing approaches, e.g. Rocchio, RFE, SVM, Random Trees, etc.

Table 4: Comparison between FKRF RBF on all data (T=1) and RFRF RBF (T=1) (MAP values).

Feature	FKRF for all data	FKRF RBF
Visual Features	34.02%	**38.23**%
Standard Audio	38.25%	**46.34**%
Text	32.37%	**35.14**%

5.4 Fisher Kernel representation on all data

We could also generate a Fisher Kernel representation by learning a GMM on *all* the data. A valid question is therefore: do we obtain good results because the Fisher Kernel representation is in general more powerful than our initial features, or are our performance improvements caused by altering the features with respect to the top n results? In the former case, we can just alter the features once offline, which would speed up computation. Yet if this is the case we would just prove that the Fisher kernel representation is more powerful than our initial features, independent of our relevance feedback settings.

To test this, we train a GMM on all the feature vectors of the whole dataset, and represent all videos as Fisher vectors with respect to this global mixture model. We use these features in the SVM RF framework and compare this with our proposed Fisher Kernel RF framework. Notice that the only differences between these two systems are on what data the GMM is learned and when the features are changed to the Fisher kernel representation.

GMM on *all* data drastically reduce the system performance. The results are presented in Table 4. It can be observed that the performance drops with 4 percents for visual features and with more than 8 percents for audio features.

We conclude that altering the data based on the top n videos is crucial for obtaining good performance. This validates our claim that the Fisher Kernel is particularly suited for use in a Relevance Feedback application.

5.5 Frame Aggregation with Fisher kernel

In the following, we will show the improvements using FK approach on RF, when we use more then one feature per video document. Because these are preliminary experiments, we used in this work only two types of visual descriptors: HoG descriptors and MPEG 7 related descriptors, that are more representative for the visual information.

For this experiment, we extract a small collection of salient frames using [27], and compute a visual feature for each frame. Because, we now have more data, we can learn more complex GMM. Therefore, we estimate that the optimal number of centroids used by the Fisher Vectors is higher than one. Indeed, Figure 4 presents the variation of MAP using a different number of Gaussian centroids. It can be observed that the best results are obtained using 6 to 10 number of centroids.

In the end, we present in Table 5 a comparison between the MAP values of previous global FKRF approach and the frame aggregation FKRF approach. The frame aggregation Fisher kernel representation for RF tends to provide better retrieval performance in all cases with more than 4 percents increase of performance (from 29.59% to 32.87% for HoG features and from 40.80% to 45.43% from MPEG 7 related

Table 3: Comparison with state of the art algorithms (mean average precision values).

Feature	Without RF	Rocchio	NB	BOOST	SVM	RF	RFE	FK Linear	FK RBF
HoG	17.18%	25.57%	24.18%	26.72%	26.49%	26.89%	27.5%	29.46%	**29.59%**
Structural	14.82%	21.96%	23.73%	23.63%	24.62%	24.69%	23.91%	**26.28%**	23.96%
MPEG 7	25.97%	30.88%	34.09%	32.55%	32.90%	36.85%	31.93%	40.50%	**40.80%**
All Visual	26.18%	32.98%	34.25%	35.99%	36.08%	**42.28%**	32.43%	41.33%	42.23 %
Standard Audio	29.26%	32.71%	34.88%	32.88%	38.58%	40.46%	44.32%	44.80%	**46.34%**
Block Based Audio	21.23%	35.39%	35.22%	39.87%	31.46%	33.41%	31.96%	**43.96%**	43.69%
Text	20.40%	32.55%	26.91%	26.93%	34.70%	34.70%	25.82%	34.84%	**35.14%**
All Features	30.29%	37.91%	39.88%	38.88%	40.93%	45.31%	44.93%	46.43%	**46.80%**

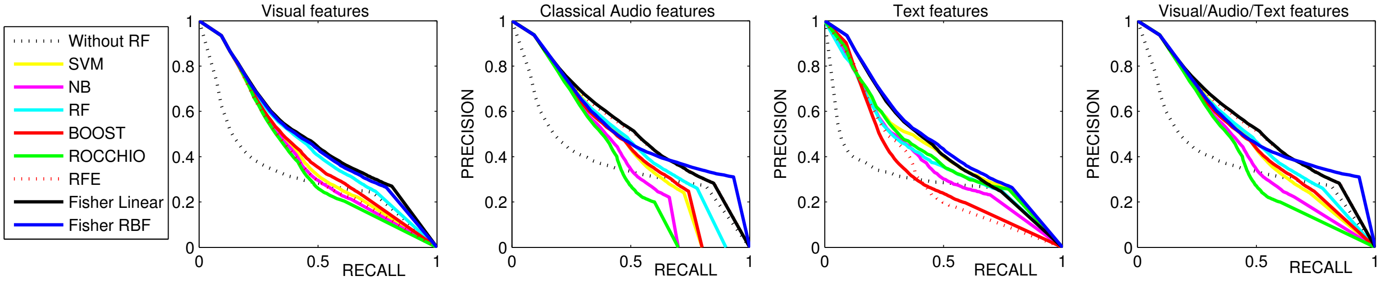

Figure 3: Precision-recall curves for different content descriptors and combinations 1 - Combination of All Visual descriptors), 2 - Standard audio features, 3 - Text descriptors and 4 - combination of all features.

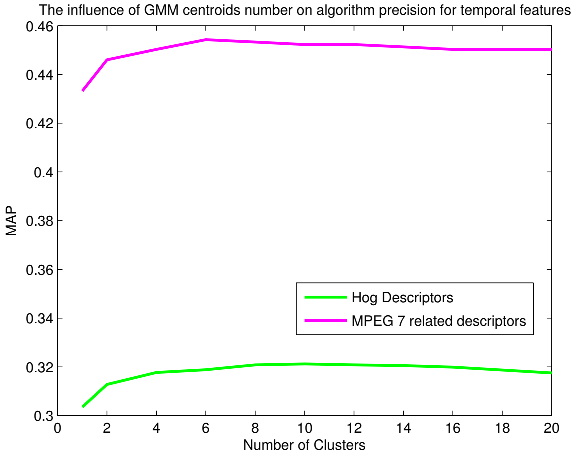

Figure 4: The influence of GMM centroids number on system performance using frame aggregation FKRF approach.

Table 5: Comparison between global FKRF and frame aggregation FKRF (MAP values).

Feature	FKRF Linear (T=1)	FKRF RBF (T=1)	Frame aggregation FKRF Linear	Frame aggregation FKRF RBF
HoG	29.46%	29.59%	32.12%	**32.87%**
MPEG 7	40.50%	40.80%	44.69%	**45.43%**

descriptors). Another interesting result is that using MPEG 7 related descriptors alone, with temporal information, we achieve similar performance to audio features.

We conclude that frame aggregation Fisher Kernels approach improves the video retrieval performance and surpasses the global Fisher Kernel approach.

5.6 Computational Efficiency

All the experiments were done on a single core of a 3.00 Ghz Intel Core Duo E8400 processor. Using Fisher Kernel in combination with Linear SVM and global video features, we generate a RF iteration in less than half of second. By aggregating all the frames with Fisher kernels, the execution time of a RF iteration is near to 2 seconds.

We conclude that this represents a reasonable waiting time for users in a real system scenario.

6. CONCLUSIONS

In this paper we have proposed a new method of relevance feedback using the Fisher Kernels. We addressed relevance feedback techniques in the context of video retrieval and discussed a new approach that combines the generative models with discriminative classifiers (SVM's) for relevance feedback problem, using Fisher Kernels theory.

Tested on a large scale video database (MediaEval 2012) and using several descriptors approaches (visual, audio and textual features) our FKRF approach improves the retrieval performance, outperforming other existing Relevance Feedback approaches, such as: Rocchio, Nearest Neighbors RF, Boost RF, SVM RF, Random Forest RF and RFE.

Additionally, we present a novel method to capture temporal information by using the Fisher Kernel to use more than one feature per video. The experiments with visual descriptors showed that using more features vectors to describe a video document, instead of only one, the performance is drastically improved, from 40.80 to 45.83 for MPEG 7 related descriptors and from 29.59% to 32.87% for HoG features. We showed that we do not need large number of clusters to train the FK framework, we achieve the best performance with only 5-10 clusters. This makes the proposed approach implementable for a real time RF approach.

In future work we will adapt the method to address a higher diversity of video categories (use of the Internet). Futhermore, we want to extend the Fisher kernel to other modalities, namely text and audio, and to use elaborated spatio-temporal features.

Acknowledgments

This work was supported by the Romanian Sectoral Operational Programme Human Resources Development 2007-2013 through the Financial Agreement $POSDRU/89/1.5/S/62557$. We also acknowledge the 2012 Genre Tagging Task of the MediaEval Multimedia Benchmark [6] for providing the test data set.

7. REFERENCES

[1] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain: "Content-based Image Retrieval at the End of the Early years", IEEE Trans. PAMI, 2000.

[2] T. Jaakkola, D. Haussler: "Exploiting generative models in discriminative classifiers", In Advances in Neural Information Processing Systems 1999.

[3] F. Perronnin, J. Sanchez, T. Mensink: "Improving the Fisher Kernel for Large-Scale Image Classification", Lecture Notes in Computer Science Volume 6314, 2010.

[4] F. Perronnin, J.A. Rodriguez-Serrano, "Fisher Kernels for Handwritten Word-spotting", 10th International Conference on Document Analysis and Recognition Pages 106-110, 2009.

[5] P. Moreno and R. Rifkin. "Using the Fisher kernel method for web audio classification", International Conference on Acoustics, Speech, and Signal Processing, pages 2417-2420, 2000.

[6] http://www.multimediaeval.org/mediaeval2012/

[7] A. F. Smeaton, P. Over, W. Kraaij: "High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements", Springer Series on Multimedia Content Analysis Theory and Applications, pp. 151-174, 2009.

[8] http://trec.nist.gov

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

[10] N. V. Nguyen, J.-M. Ogier, S. Tabbone, A. Boucher: "Text Retrieval Relevance Feedback Techniques for Bag-of-Words Model in CBIR", ICMLPR, 2009.

[11] Y. Rui, T. S. Huang, M. Ortega, M. Mehrotra, S. Beckman: "Relevance feedback: a power tool for interactive content-based image retrieval", IEEE Transactions on Circuits and Video Technology, 1998.

[12] S. Liang, Z. Sun: "Sketch retrieval and relevance feedback with biased SVM classification", Pattern Recognition Letters, 29, pp. 1733-1741, 2008.

[13] G. Giacinto: "A Nearest-Neighbor Approach to Relevance Feedback in Content-Based Image Retrieval", ACM Confenference on Image and Video Retrieval, 2007.

[14] J. Yu, Y. Lu, Y. Xu, N. Sebe, Q. Tian: "Integrating Relevance Feedback in Boosting for Content-Based Image Retrieval", ASSP, 2007.

[15] Y. Wu, A. Zhang: "Interactive pattern analysis for relevance feedback in multimedia information retrieval", Multimedia Systems, 10(1), pp. 41-55, 2004.

[16] L. Yuanhua Lv, C. Zhai: "Adaptive Relevance Feedback in Information Retrieval", Information and Knowledge Management Conference, 2009.

[17] W. Bian, D. Tao: "Biased discriminant euclidean embedding for content-based image retrieval", IEEE Trans. Image Process., 545-554, 2010.

[18] D. Tao, X. Li, S. Maybank: "Negative samples analysis in relevance feedback" IEEE Trans. Knowl. Data Eng., 568-580, 2010.

[19] A. G. Hauptmann, M. G. Christel, and R. Yan: "Video retrieval based on semantic concepts", Proceedings of the IEEE, vol. 96, pp. 602-622, 2008.

[20] T. Mei, B. Yang, X. Hua, S. Li: "Contextual Video Recommendation by Multimodal Relevance and User Feedback", Information Systems (TOIS), 2011.

[21] B. Ionescu, K. Seyerlehner, I. Mironica, C. Vertan, P. Lambert: "An Audio-Visual Approach to Web Video Categorization", MTAP, 2012.

[22] I. Mironica, B. Ionescu, C. Vertan: "The influence of the similarity measure to relevance feedback", in Proceedings of the European Signal Processing Conference, Eusipco 2012.

[23] S.H. Cha: "Comprehensive Survey on Distance/Similarity Measures Between Probability Density Functions", Int. Journal of Mathematical Models and Methods in Applied Sciences, 2007.

[24] Y. Rubner, C. Tomasi, L. J. Guibas: "A Metric for Distributions with Applications to Image Databases", European Conference on Computer Vision, 1998.

[25] E. Deza, M.M. Deza: "Dictionary of Distances", Elsevier Science, 1st edition, 2006.

[26] M. Hatzigiorgaki, A. N. Skodras: "Compressed Domain Image Retrieval: A Comparative Study of Similarity Metrics", SPIE Visual Communications and Image Processing, vol. 5150, 2003.

[27] P. Kelm, S. Schmiedeke, T. Sikora, "Feature-based video key frame extraction for low quality video sequences", WIAMIS, 2009.

[28] K. Seyerlehner, M. Schedl, T. Pohle, P. Knees: "Using Block Level Features for Genre Classification, Tag Classification and Music Similarity Estimation", Music Information Retrieval Evaluation eXchange, 2010.

[29] C. Liu, L. Xie, H. Meng: "Classification of music and speech in mandarin news broadcasts", Conf. on Machine Speech Communication 2007.

[30] Yaafe core features, http://yaafe.sourceforge.net/

[31] T. Sikora: "The MPEG-7 Visual Standard for Content Description - An Overview", IEEE Transactions on Circuits and Systems for Video Technology, 2001.

[32] O. Ludwig, D. Delgado, V. Goncalves, U. Nunes: "Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection", IEEE Int. Conference On Intelligent Transportation Systems, 1, pp. 432-437, 2009.

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

[34] S. Nowak, M. Huiskes: "New strategies for image annotation: Overview of the photo annotation task at ImageClef 2010", In the Working Notes of CLEF 2010.

[35] L. Lamel, J.-L. Gauvain: "Speech Processing for Audio Indexing", Int. Conf. on Natural Language Processing, LNCS, 5221, pp. 4-15, Springer Verlag, 2008.