Multimed Tools Appl (2010) 47:631–662
DOI 10.1007/s11042-009-0340-6

Simulated evaluation of faceted browsing based on feature selection

Frank Hopfgartner · Thierry Urruty · Pablo Bermejo Lopez · Robert Villa · Joemon M. Jose

Published online: 4 August 2009

Abstract In this paper we explore the limitations of facet based browsing which uses sub-needs of an information need for querying and organising the search process in video retrieval. The underlying assumption of this approach is that the search effectiveness will be enhanced if such an approach is employed for interactive video retrieval using textual and visual features. We explore the performance bounds of a faceted system by carrying out a simulated user evaluation on TRECVid data sets, and also on the logs of a prior user experiment with the system. We first present a methodology to reduce the dimensionality of features by selecting the most important ones. Then, we discuss the simulated evaluation strategies employed in our evaluation and the effect on the use of both textual and visual features. Facets created by users are simulated by clustering video shots using textual and visual features. The experimental results of our study demonstrate that the faceted browser can potentially improve the search effectiveness.

Keywords Video retrieval · Feature selection · Clustering · Log file analysis

F. Hopfgartner (✉) · R. Villa · J. M. Jose
Department of Computing Science, University of Glasgow, Glasgow, UK
e-mail: hopfgarf@dcs.gla.ac.uk

R. Villa
e-mail: villar@dcs.gla.ac.uk

J. M. Jose
e-mail: jj@dcs.gla.ac.uk

T. Urruty
University of Lille 1, Villeneuve d'Ascq, France
e-mail: thierry.urruty@lifl.fr

P. B. Lopez
University of Castilla-La Mancha, Albacete, Spain
e-mail: pbermejo@dci.uclm.es

1 Introduction

The increasing popularity of online video services, such as YouTube[1] and DailyMotion[2], has led to the need for novel methods for searching video databases. The performance of video retrieval algorithms to date is poor compared to widely employed text retrieval algorithms. In addition, efforts aiming at improving video retrieval face the problem of the "Semantic Gap" [23]. This is the large difference between low-level features which can typically be extracted automatically from image, video and audio data for representation/indexing, and the semantic concepts which users typically use to search. However, these deficiencies can potentially be addressed by empowering users with more effective retrieval interfaces which allow users to explore, browse, and organise their search tasks.

Current video retrieval approaches, in particular the retrieval systems evaluated in TRECVid[3] [22] model retrieval in a "one result list only" approach, which assumes the user is focused on one particular search issue. An example of this type of search task is: "Find shots of Condoleezza Rice". These tasks are useful in benchmarking various retrieval algorithms as shown in the TRECVid evaluation experiments, however, they are not representative of real world video information seeking tasks. For example, a researcher or journalist at a broadcasting station who is searching for material to use in the production of an item for the evening news, may be interested in highlighting the achievements of multiple swimmers at the 2008 Olympic Games in Beijing. However, as they progress through the search task, they may become interested in highlighting other issues, such as preparatory issues related to the performance of Michael Phelps, or to highlight the need for more governmental support in the development of future swimmers. Current retrieval systems and approaches fail to provide any support for such broad, multi-faceted tasks. In a faceted retrieval system, one may search for information about various aspects of the underlying information need without interrupting the current search session.

One important problem inherent to multimedia information retrieval is the use of low level visual features to retrieve relevant documents. Retrieval using low-level features faces two major problems. The first is the well known "curse of dimensionality", which has been studied extensively, i.e. in [31]. To overcome this problem, solutions have been proposed in the field of multidimensional indexing structures, involving the creation of structures which allow efficient access to multimedia databases [21, 26]. Other researches have proposed the use of dimensionality reduction by selecting the most appropriate dimensions [6, 15]. The second main problem in multimedia information retrieval is the unsatisfactory performance of video retrieval systems due to the semantic gap. High level feature extraction or annotation techniques are application dependant; the results of TRECVid experiments to date show the inadequacy of content based search systems and also the limited effectiveness of high level feature extraction systems [22].

[1] www.youtube.com

[2] www.dailymotion.com

[3] TRECVid is a large scale evaluation campaign aiming at research problems related with video data.

So far, no simulated evaluation of faceted browsing exists. One motivation of this work is to provide a methodology that demonstrates the potential benefits of a faceted search and browsing system. In this paper, we first introduce an approach to reduce the dimensionality of low level visual features to overcome the "curse of dimensionality", hence decreasing the query processing time of content based retrieval systems. This will enable the on the fly querying with different low level features possible, which is useful in proposing different facets of a search task, where multiple searches must be carried out in parallel with speed. The presented experiments are based on an exhaustive analysis of visual features for the TRECVid 2006 corpus. Secondly, we study the concept of facet-based retrieval as an aid to bridging the semantic gap. We propose a novel simulation methodology to evaluate the effectiveness of faceted browsing in which we simulate users creating new facets in an interface. We then discuss the different strategies used in our simulation. Finally, we support our results by exploiting the log files which have been generated in a previous user study. Our work is based on an interactive video retrieval system and evaluation presented in Villa et al. [28].

The rest of the paper is organised as follows. Section 2 provides an overview of existing research related to this work. In Section 3, we present our methodology for selecting features to enable dimensionality reduction. In Section 4, we introduce a novel simulated evaluation methodology for faceted browsing which iteratively clusters retrieval results based on their visual and textual features. The results of this approach indicates that faceted browsing can be used to improve retrieval effectiveness. Subsequently, we analyse user logs from a previous user study [28] to verify our results in Section 5. In Section 6, we discuss the results of our various approaches.

2 Background

In this section we first discuss prior approaches in assisting the user in retrieving different facets of a topic. Then, Section 2.2 gives details about the faceted interface we used. Finally, in Section 2.3, we argue for the use of user simulation to evaluate our approach.

2.1 Facet-based retrieval

Within the TREC-5 interactive track, a major information retrieval evaluation campaign, the term "aspect" is used and defined as "roughly one of many possible answers to a question which the topic in effect posed" [17]. Similar topics were used in TREC-7 and TREC-8, indicating that retrieving different aspects is considered to be an important research question. For example, Topic 408i from [9] has the description "What tropical storms (hurricanes and typhoons) have caused property damage and/or loss of life?", and in its associated instances section asks the user to "[...] find as many different storms of the sort described above as you can [...]".

Harper and Kelly [8] use the aspectual search topics provided within TREC-8 to evaluate an information retrieval interface which provides the user with facilities for the organisation of retrieval results within different *piles*. Each pile can be used as a source of relevance information for executing new queries. Kerne et al. [13] introduce an interface which allows users to combine image and text summaries in order to promote idea generation and discovery. While this system does provide space for users to organise information, the focus is more general, not being solely intended for search tasks.

Methods applied in the text retrieval cannot easily be adapted to the multimedia domain, largely due to the semantic gap [23], while recommending similar videos based on text queries is challenging because most videos are not annotated [7]. Content-based retrieval models ease this problem by relying on low-level features which can be extracted from the videos. One example is the EGO system [25], which provides media professionals with a workspace in which to organise their information needs, and provides retrieval based on low-level visual features. A similar system is ImageGrouper [16], which allows users to group query examples in order to improve the performance of content-based image retrieval. However, in this approach, the creation of groups is a separate process from the process of search.

Villa et al. [29] propose an alternative search environment by introducing a *faceted browser interface* which supports the creation of multiple search panels. Their study suggests that providing users with the facility to re-arrange retrieved results between panels aids the user, for broad and complex search tasks. As our work is based on their interface, which is briefly presented in the next section, we follow their terminology of calling each "aspect" a "facet" of a search task.

2.2 A facet-based video retrieval system

In this section, we introduce the implementation of a facet-based video retrieval system. Further details can be found in [29]. As in many retrieval systems, it is divided into a frontend search interface, and a backend system which implements the underlying retrieval functionality.

The faceted interface shown in Fig. 1 is split into one or more vertical panels, each panel representing a single facet of a larger task. Each panel can be used to enter different queries, and will display the corresponding query's search results. When initially started, a single panel is displayed; new panels can be created using the "Add" button on the top left of the screen. Following the marked numbers on Fig. 1, each panel contains: (1) name for the panel which can be provided by the user; (2) delete icon which deletes the entire panel; (3) a key shot intended to be used as a visual exemplar for the panel, selected by the user; (4) left and right arrows, which will move the panel left or right within the overall sequence; (5) search box and button, allowing the user to enter a textual query and start a search; (6) pull down list of queries already carried out in that panel; (7) list of relevant shots, as selected by the user for that panel; (8) a list of search results; and (9) is the scroll bar appearing when the number of facets is too high.

The interface makes extensive use of drag and drop. Shots on the search result list can be dragged and dropped onto the relevant shots area, which will add the shot to the facet's list of relevant shots. There is no restriction on what facet a result can be dragged onto, therefore it is possible to drag a result from one facet directly onto the relevant list of a different facet. Relevant shots can also be dragged and dropped between the different facet list of relevant shots, allowing the re-organisation of material across the different facets. Relevant shots can be removed

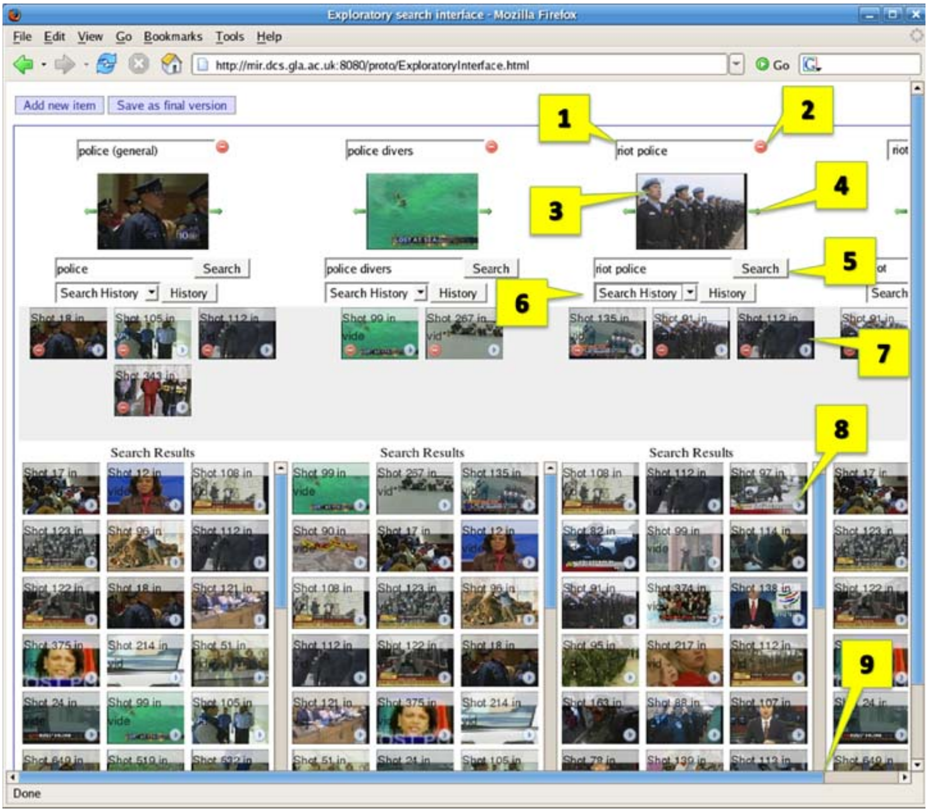


Fig. 1 Screenshot of the facet browsing interface, the numbers referred to in the text

from the relevance lists using a delete button given on the bottom left of each shot's keyframe.

The backend indexes video shots based on text associated with each shot using a conventional information retrieval system and also based on the low level visual features of the keyframe chosen to best describe the shot. BM25 [19] is used to rank results for text query and the Euclidean distance ranked the results coming from the visual query. A user can type a query or choose a keyframe to create a query-by-example.

2.3 Evaluation methodology

Most interactive video retrieval systems are evaluated in laboratory based user experiments. This methodology, based on the Cranfield evaluation methodology, is inadequate to evaluate interactive systems [10]. User-centred evaluation schemes are very helpful in getting valuable data on the behaviour of interactive search systems, however, they are expensive in terms of time and money, and the repeatability of such experiments is questionable. It is almost impossible to test all the variables involved in an interaction and hence compromises are required on many aspects of testing. Furthermore, such a methodology is inadequate in benchmarking various underlying adaptive retrieval algorithms. An alternative way of evaluating such systems is the use of simulations.

Finin [3] introduced one of the first user simulation modelling approaches. The "General User Modelling System" (GUMS) allowed software developers to test their systems by feeding them with simple stereotype user behaviour. White et al. [33] proposed a simulation-based approach to evaluate the performance of implicit indicators in textual retrieval. They simulated user actions such as viewing relevant documents, which were expected to improve the retrieval effectiveness. In the simulation-based evaluation methodology, actions that a real user may take are assumed and used to influence further retrieval results. Hopfgartner and Jose [10] employed a simulated evaluation methodology which simulated users interacting with state-of-the-art video retrieval systems. They argue that a simulation can be seen as a pre-implementation method which will give further opportunity to develop appropriate systems and subsequent user-centred evaluations. However, this approach to evaluation is not mature, and there is a need to develop techniques to simulate user behaviours which are appropriate for the system under consideration.

In the following sections, we present a methodology for the selection of visual features. Reducing the dimensionality of data results in a faster query processing time. Then, in Section 4, we introduce our proposal to simulate user behaviour on a faceted browser. Finally, Section 5 presents our approach to verify the outcome of our simulated evaluation, by exploiting the logfiles of a user study.

3 Feature selection

A major challenge in Multimedia Information Retrieval is to judge the relevance of a document to a given query. This can be computed using "visual features" of an image or a keyframe of a video shot. These visual features might have a high dimensionality and/or some others might offer a very low predictive power. Moreover, the underlying database, like other collections, may suffer a common problem in multimedia information retrieval: that of a high imbalance between relevant and non relevant documents for each search topic, a problem we denote as the *skewed data* problem. In this section we deal with these problems and present our work to solve them. First, we explain our methodology for solving the problem of skewed data. Then we propose a Feature Subset Selection methodology on visual features to reduce the features' dimensionality. Finally, we perform an exhaustive search to identify the best combinations of visual features and show the potential for the speeding up of the retrieval by feature selection.

3.1 Balance of training data

In this section, we approach a retrieval task as a supervised classification problem with a binary class attribute ("relevant" and "non-relevant"). Classified documents are a set of instances, each one representing a shot using a visual feature, such as Colour Layout. Formally the problem can be established from a set of instances $C_{train} = \{(s_i, l_i), \forall i\}$, such that $s_i \in S$ is the instance which corresponds to the shot i of the set of shots S, $l_i \in L$ corresponds to the value of class attribute that contains the shot s_i and $L = \{\text{relevant, non-relevant}\}$ is the set of possible values for the class attribute. The goal is to build a classifier $c : S \to L$ to solve the prediction of shots' relevance; that is, the value of the class attribute for each instance.

A well known problem when performing a classification on a real corpus is the lack of balance between each class of the training set. Classifiers such as Naïve Bayes might overfit the learnt parameters. For non parametric classifiers based on neighborhood, unbalanced classes result in some *invasion* in the vectorial space. This phenomenon provides incorrect classification for documents whose correct class appears just a few times in the training set.

Regarding a dataset of instances containing v possible values for class attribute, and M being the number of instances belonging to the most frequent class and m the number of instances belonging to the least frequent class, methods to balance this dataset can be classified as:

1. *Sample until balanced.* New instances are sampled and added to the dataset until it contains M instances for each class. There are several ways to sample instances, such as just copying the existing ones or sampling new ones from a learnt distribution or property from the instances belonging to each class.
2. *Remove until balanced.* This method consists of removing instances until the dataset contains m instances for each dataset. Instances might be removed by merging them or by deleting them from the dataset based on some criterion.
3. *Sample a whole new set.* The number of instances for each class is set to P, then the distribution is learnt from instances belonging to each class. Finally, P instances are sampled for each class using the previously learnt distribution (as done in [2]).

Our experiments are based on the TRECVid 2006 data collection. This corpus consists of approx. 160 hours of television news video in English, Arabic and Chinese language which were recorded in late 2005. The dataset also includes the output of an automatic speech recognition system, the output of a machine translation system (Arabic and Chinese to English) and the master shot reference. Each shot is considered as a separate document and is represented by text from the speech transcript. In the collection, we have 79484 shots and 15.89 terms on average per shot, with 31583 shots without annotation. We use the set of 24 topics contained in the data collection. Each topic contains a query of several keywords and relevant keyframes and also a judgment list of 60 to 775 relevant documents. In our experiments, we used five low level features: Colour Layout (12 dim.), Dominant Colour (15 dim.), Contour Shape (130 dim.), Homogeneous Texture (62 dim.) and Edge Histogram (80 dim.). We denoted them *CL*, *DC*, *CS*, *HT* and *EH* respectively.

The video shots of TRECVid 2006 corpus are the instances for training and classification, from which only an average of 300 shots are relevant for each search topic. Therefore, we have a huge and highly skewed set of shots for which we must learn and predict their relevance. Our evaluation methodology is to perform a 10 × 10 cross-fold validation (10 × 10CV). Since the test sets cannot be modified and splits are made randomly in each run, a balance of training sets needs to be made at execution time. Since the database contains about 80000 instances, using a 10 × 10CV is time consuming so the balancing methodology should be as light as possible.

In our work, we choose to use a *remove until balanced* approach to balance training sets without adding extra load to our 10 × 10CV. We denote α the degree of balance. Let N be the difference between the number of non-relevant and relevant documents in the training set, then we define α as the percentage of N non-relevant instances to be removed. Thus, when $\alpha = 100$ we transform the training set into a set with the same number of relevant and non-relevant instances. If we set $\alpha = 0$, no change is made to the training set.

We performed the 10×10CV using different values of α to find out its most appropriate value for relevant prediction with respect to the corpus. We present results obtained from evaluations using three common measures in classification problems: *precision*, *recall* and F_1 measure, the harmonic mean between precision and recall, giving the same importance to each of them [27].

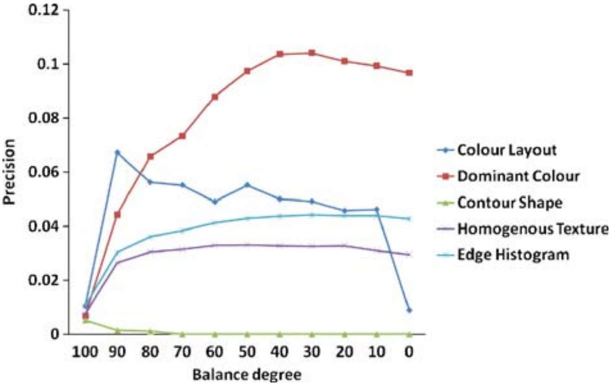
First, a 10×10CV has been run over TRECVid 2006 for each visual feature and for each of the 24 search topics in TRECVid 2006, using a balance degree of $\alpha = 100$. This cross-fold validation was performed using four different classifiers: Naïve Bayes, AODE [30], Support Vector Machines and k-Nearest Neighbour. The probabilistic classifier AODE is the best option to find a compromise between speed and performance. Then, another 10×10CV was run using AODE for each visual feature and each of the 24 search topics in TRECVid 2006, using balance degrees from 0 to 100. Results for each topic are averaged and are shown in Figs. 2, 3 and 4 for eleven different values of α and for five different low level visual features used to represent shots in database. The two graphics in Figs. 2 and 3 show the common behavior of *precision* and *recall*: as one increases the other decreases.

These figures show high precision values for the Dominant Colour (DC) and Colour Layout (CL) visual features. This is mainly due to outlier search topics which are related to sports. Their performance is much lower for any other topic. Thus, we conclude that, in general, the three best performing visual features for all topics are Dominant Colour, Homogeneous Texture and Edge Histogram. To find a good breakeven we computed the F_1 measure shown in Fig. 4, thus we can decide to fix the balance degree $\alpha = 50$ for our next experiments as this value is on average the best F_1 measure value.

3.2 Feature subset selection

In order to improve the prediction power of the classifier, two types of selection have been performed: visual feature dimensionality reduction, presented in this section;

Fig. 2 Precision for relevant shots prediction using AODE classifier



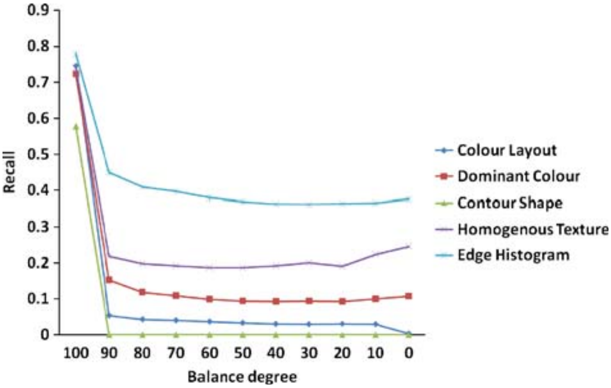


Fig. 3 Recall for relevant shots prediction using AODE classifier

and visual feature selection performing an exhaustive search, explained in the next section. The quality of the used set of features is of great importance for the classifier to achieve a good performance [1]. This performance depends on the individual relevance of each feature with respect to the class, the relationship among features and the existence of features which affect negatively the classifier. It is possible to improve the quality of the available features by performing:

- **Feature Subset Selection**. This is a widely studied task [6, 15] in data mining, and it consists of reducing the set of available features by selecting the most relevant ones using filter metrics (statistical, distances, etc.) or a wrapper (goodness of the classifier).
- **Feature Construction**. New features with a higher quality are obtained by computing some relation or statistic from original features as area, ratio, differences,etc. This task is known as *feature construction* [11, 14], and we do not deal with it in this work.

Feature Subset Selection (FSS) is the process of identifying the input variables which are relevant to a particular learning (or data mining) problem. Though FSS is of

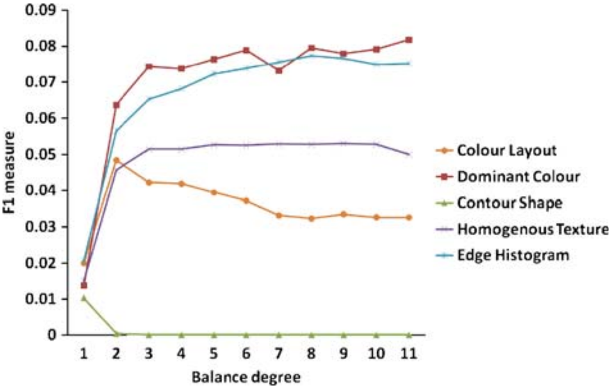


Fig. 4 F_1 measure prediction using AODE classifier

interest in both supervised and unsupervised data mining, we focus on supervised learning, and concretely in the classification task. That is, projecting information the retrieval problem in a classification task, we consider the existence of a distinguished variable (*the class*) whose value is known in the dataset instances. Classification oriented FSS carries out the task of removing most irrelevant and redundant features from the data with respect to the class. This process helps to improve the performance of the learnt models by:

- Alleviating the effect of the "curse of dimensionality" problem;
- Increasing the generalisation power;
- Speeding up the learning and inference process;
- Improving model interpretability.

Unlike other dimensionality reduction techniques (e.g. principal component analysis), FSS does not alter the original representation, so it preserves the original semantics of the variables, helping domain experts to acquire better understanding about their data by informing them of which are the important features and how they are related with the class. In supervised learning, FSS algorithms can be (roughly) classified into three classes: (1) embedded methods; (2) filter methods; and, (3) wrapper methods. By embedded methods we refer to algorithms such as C4.5 [18], that implicitly use the subset of variables they need. Filter techniques evaluate the goodness of an attribute or set of attributes by using only intrinsic properties of the data (e.g. statistical or information-based measures). Filter techniques have the advantage of being fast and general, in the sense that the resulting subset is not biased in favour of a concrete classifier. On the other hand wrapper algorithms use a classifier (usually the one to be used later) in order to assess the quality of a given attribute subset. Wrapper algorithms have the advantage of achieving a greater accuracy than filters but with the drawback of being (by far) more time consuming and obtaining an attribute subset that is biased toward the used classifier, although in the literature we can find some attempts to alleviate this problem [4]. However, wrapper methods have the disadvantage of being time consuming and biasing the result (with respect to the classifier used) stronger than filter methods.

We are tackling the information retrieval task as a classification problem and, as such, we can perform a dimensionality reduction for each visual feature. A feature can be regarded as an observation from a sample, and from that point of view it would be interesting to have as many observations as possible. However, a large array of observations might contain a lot of noise which leads to wrong conclusions. Besides this, TRECVid 2006 is a database with a huge number of samples so no long visual features should be needed to feed the classifier. Moreover, when studying the visual feature instanciations which describe shots in TRECVid 2006, we find that some dimensions are always set to 0. So our hypothesis is that a feature subset selection might be helpful to improve the classifier's performance in terms of time and/or F_1 measure for the TrecVid2006 corpus. In [20], the authors perform selection using a Feature Vector Reduction process on two COREL collections. Although results are good, they fix the reduced vector to represent color and texture visual features without explaining why. In this work, we do not previously choose any visual feature but perform visual feature dimensionality reduction and an exhaustive visual features selection.

In the following, we denote "visual feature", as one of the low level visual features previously described; and we denote "feature" when we refer to one of the dimensions inside a visual feature. Thus, the visual feature *Colour Layout* is composed of 10 features; that is, it has 10 dimensions.

Since wrapper methods bias the results toward the wrapper classifier and our goal is to apply the results to information retrieval systems, we decided to use a filter metric to perform feature selection; TRECVid 2006 is such a huge corpus that a filter metric is needed. In [32], the authors present a mathematical study from which they conclude that information-based metrics as *Information Gain* (IG) are biased, favoring the selection of nominal attributes which have a higher number of states. However, a more modern study [5] performed experiments over a huge workbench and concluded that the "Information Gain metric is a decent choice if one's goal is precision", which is our case since information retrieval system aim for that performance measure. Forman's work compares different information-based and statistical metrics (including chi-square), and concludes that "under low skew, IG performs best and eventually reaches the performance of using all features". Since we balance our training sets, we have a very low skew. So, based on this work, we select Information Gain as the metric used for feature selection.

For each visual feature, the IG value for each feature with respect to the class is used to create a ranking to know which indexes of the vector describing each visual feature is more relevant with respect to the class. Then, the best percentage P of features in the ranking is projected over the database and classification is performed to compute how good this new subset is. This classification is performed as described in Section 3.1, and training sets are balanced setting $\alpha = 50$ as was previously computed to reflect the best level of balance. Several values for P have been tested and precision, recall and F_1 measure values have been computed. F_1 measure values are shown in Fig. 5.

These results confirm our expectation: a fine feature subset selection can be done for visual features. Keeping just the best 40%, 50% and 60% of features ordered by their IG with respect to the class makes the classifier have a slight loss in predictive power (based on precision for relevant documents) while reducing by half the dimensions of visual features. Information retrieval systems and especially

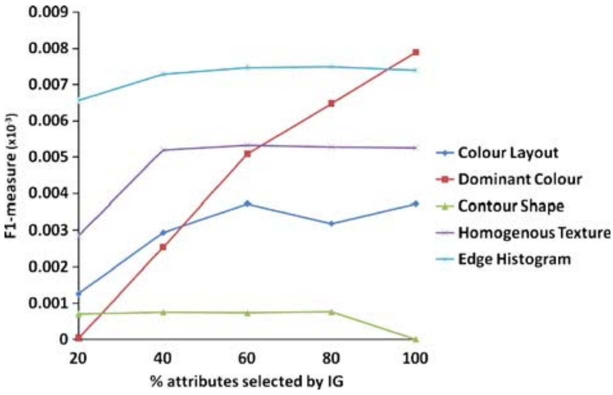


Fig. 5 F_1 measure over different values for P using AODE classifier

indexing structures could benefit from this reduction of dimensionality by achieving a faster response to user's queries without losing quality in their final list of suggested documents.

3.3 Visual feature selection

Our hypothesis here is that the combination of two or more visual features might improve the performance of the classifier. Since we are working with 5 visual features, the search space consists of 31 possible combinations. Thus, although it is a time consuming task, we perform an exhaustive search to find out which combination of visual features makes our classifier work better.

Figure 6 shows the values for F_1 measure averaged over all the 24 TRECVid 2006 topics. These results were expected. Indeed they reflect the previous results showing the potential of combining Dominant Colour, Homogeneous Texture and Edge Histogram as the best combination of three features and then Dominant Colour and Colour Layout as the best combination of two visual features. They also demonstrate that the Contour Shape visual feature is not useful for this collection. These results show also that combining visual features tends to improve the performance of classifier, however, this also means an increase in computational load.

With the best combinations, we have performed a Feature Subset Selection (same methodology as used in Section 3.2) to check if we can keep their good performance and decrease their dimensionality. So we selected the 40% and 60% in an IG-ranked list of the features belonging to each combination. The results show that the F_1 measures values do not decrease for these combinations of features while their dimensionality can be also reduced by half.

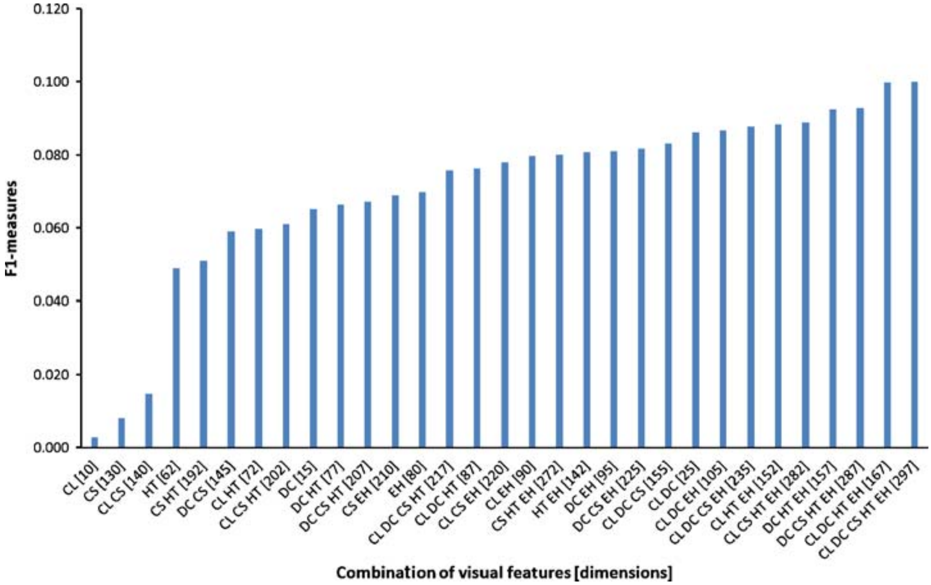


Fig. 6 F_1 measure for all possible combinations of visual features

3.4 Discussion on feature selection

In this section we have dealt with three problems:

1. Skewed data. Our experiments help us to fix the optimal degree of balance in the training set between "relevant" and "non-relevant" instances for the used classifier. The balancing strategy is fast and can be done without affecting the evaluation time.
2. Feature subset selection. We successfully reduced the dimensionality of visual features without decreasing the performance of classifier, finding out that we can remove up to 60% of the worst (based on IG) features, for each visual feature.
3. Visual features combination. We have performed an exhaustive search to find which combination of 5 visual features performs best (in terms of F_1 measure values) to predict the relevance of documents. We found that best combinations of visual features are mostly based on Edge Histogram, Dominant Colour and Homogeneous Texture. Since these combinations still have a high dimensionality, we performed the Feature Subset Selection based on IG-ranking finding that we could reduce the dimensionality without decreasing the effectiveness of the retrieval.

4 Simulating user behaviour for the evaluation of faceted browsing

As described in the background section, facet-based retrieval has rarely been studied, especially for multimedia data. Our objective is to study the bounds of the proposed faceted browser. We therefore employ a novel simulated evaluation methodology which assumes a user is acting on the faceted system. If such a user is available, he or she will do a set of actions that, in their opinion, will increase the chance of retrieving more relevant documents. One way of doing this is to select relevant videos. By using a test collection like TRECVid, we will be able to use the available relevance judgements for the simulation.

In this section, we first introduce our methodology for simulating users creating facets, in order to evaluate faceted browsing. The idea is to make use of clustering to create groups of similar objects. The clusters are assumed to be the facets of a user's search needs and are hence used in the simulation. First, we explain the mechanisms of our algorithm using an iterative clustering technique, then we detail our experimental setup and the various simulations we made before finally discussing the experiment results.

4.1 Iterative clustering methodology

The main goal of our facet-based interface is to help the user to create a complex query with separated and structured views of different sub-queries. Our iterative clustering approach mainly aims to simulate the user in his or her search task. Clusters of our algorithm are assumed to be the facets a real user may create during a search process. A user's first query has a high probability of being general, with the retrieved set of results containing different semantic topics, e.g. if the query contains “sport” as a keyword, the system will retrieve results of different sports and also other results such as people commenting on a match. Hence, we may obtain a set of more coherent facets for the user, e.g. a facet on “football” or “basketball” and another facet on “people commentaries”. Figure 7 shows an overview of our approach integrated within the facet-based interface.

On the top left of the figure is the starting facet, where the user launches a query-by-text or a query-by-example via the user interface (Facet 1). This query may contain text or/and visual features from images. The retrieval backend returns a list of results displayed in the first facet of the interface. Our iterative clustering algorithm starts at this step (the coloured parts of the figure).

First, we cluster the retrieved results using textual and visual features. We assume that the top k clusters form the k facets of a user’s need and use them to create more specific queries. These queries will then be used to automatically propose new sets of results in new facets. Finally, the iterative clustering process is used to find new facets and refine the queries and consequently the retrieved results. The iterative process allows the display of the k result lists as new facets on the interface, or the launching of a new clustering call on each result list; we discuss this process further below.

We choose to use an unsupervised agglomerative hierarchical clustering and the single link method [12]. Let C, D be two clusters, So_C, So_D the respective set of objects of clusters C and D, the single linkage equation between C and D is given by the following formulas:

– for visual features of images representing video shots we use:

$$D_{SL}(C, D) = \text{Min}\{d(i, j), \forall i \in So_C \text{ and } \forall j \in So_D\} \tag{1}$$

where $d(i, j)$ is the Euclidean distance;

– for text queries, we use:

$$D_{SL}(C, D) = \text{Max}\{d(i, j), \forall i \in So_C \text{ and } \forall j \in So_D\} \tag{2}$$

where $d(i, j)$ is the number of common annotation keywords between two documents.

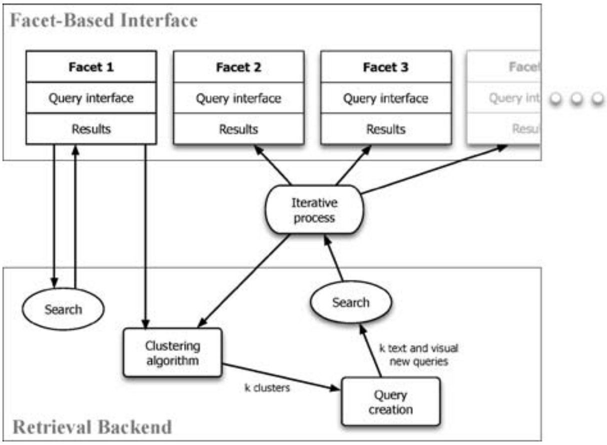


Fig. 7 Mechanisms of our iterative clustering proposal

The output of a hierarchical clustering algorithm is a dendogram. The number of clusters wanted is a parameter of our algorithm, which is used to create the k clusters. We then create a new query for each cluster. For visual features, we choose the medoid (the object closest to the centroid) of the cluster to create the new visual query. The new text query is based on the most common keywords annotating the cluster. Several combinations and number of keywords have been tested to create the new text query; we present the different results of our experiments in Section 4.3. A new search is launched to retrieve k new sets of results corresponding to the k new queries which are displayed in different facets of the interface.

We apply clustering on the initial results of the query above. The resulting clusters are used for identifying new facets and subsequently new queries are generated, as explained above. The process is repeated iteratively to identify new facets and hence new queries. This iteration can be done in two ways. The first method is completely automatic: results from the first clustering call are directly clustered again to add more precision to the queries. This requires a *number of iterative calls* parameter, denoted N_{ic}. The number of facets N_f that are proposed to the user at the end of the iterative phase is equal to $N_f = k^{N_{ic}}$, so both parameters k and N_{ic} should be low. A "facet waiting queue" may be required if these parameters are too high. The second method requires interactions with the user. At the end of the first clustering phase, new results are displayed in the facet-based interface. Then, for each facet, we simulate the user's actions, e.g., he may choose to delete the facet, to keep it, or to launch a new clustering call. Such actions are simulated based on the number of relevant documents in each cluster. For example, clusters with more relevant documents are used as a facet. This "user-simulated interactive" method has some advantages: first it is better adapted to the free space of the interface as the user may delete non relevant facets before each new call; and finally, it does not require the N_{ic} parameter.

In the following sections, we present the experimental setup and our various experiments which lead to the main conclusion that faceted browsing can improve the effectiveness of the retrieval.

4.2 Experiment setup

Our different experiments are based on the TRECVid 2006 dataset. As we introduce a novel simulated evaluation for faceted browsing, benchmarking it with systems introduces within TRECVid is not possible. Thus, we use a baseline system to evaluate the potential of our approach. As in previous simulation based approaches, retrieval precision is reported. Indeed, we compute iteratively the precision values of the clusters and automatically select the k best sets of results for the next iterative call. These are the sets of results that have the highest precision, as our goal is to simulate the actions of a user creating new facets. For our experiments, we set $k = 3$, because the list of retrieved results contains only 100 results, which is too small to perform a clustering for higher k values.

In most of the experiments presented in the following sections, we compare our iterative clustering approach with a baseline run. For each topic, this baseline run uses the topic description as a list of keywords to retrieve a baseline list of results and a precision value is computed using the relevant list for the topic.

Table 1 Results using only visual features queries

Visual features	–	=	+
Dominant colour	14	10	0
Colour layout	14	6	4
Texture	12	6	6
Edge histogram	11	5	8
Contour shape	17	3	4
Average	13.6	6	4.4

4.3 Results

In the first step of our experiments, we simulate users creating new facets in the faceted browser. First we show the results based on only one visual feature. Then we expand the query by adding more keywords to the initial text query.

Experiment on visual queries A visual query is based on the visual features of one or several images. For this set of experiments, we separately used five different low-level features extracted using the Mpeg-7 library[4]: dominant colour, texture, colour layout, contour shape and edge histogram. As the precision values of the baseline run are based on text queries, they are not used in this set of experiments. We record the evolving precision values for various steps of our iterative clustering approach based on visual feature queries only.

Table 1 presents the results of our iterative clustering algorithm. For each topic and each feature, we present our results in three different categories:

- the precision value of the best results decreases more than 2%, denoted "–";
- the precision value of the best results is almost stable, denoted "=";
- the precision value of the best results increases more than 2%, denoted "+";

As an example, the iterative clustering results based on the texture features increase the precision of results for six topics (out of 24). However, for half of the topics the precision decreases. The conclusion we can draw from Table 1 is that visual features are not reliable for every query. However, for some of the topics, they are useful and improve the precision of the retrieved results. This corroborates with the findings presented by Smeaton et al. [22].

In Fig. 8, we show three examples of decreasing, stable and increasing precision results with respect to the number of iterative clustering calls for different visual features combined with the initial text query. We present the evolution of the precision using dominant colour visual feature on topic 181, denoted "DC 181", colour layout visual feature on topic 195, denoted "CL 195" and edge histogram on topic 182, denoted "EH 182".

Experiment on text queries For a further analysis of the introduced clustering methodology, we evaluate our iterative clustering algorithm on text queries by presenting a set of experiments using query expansion. One, two or three new keywords are added to the existing keywords of the initial query, denoted "add 1","add 2" and "add 3", respectively. The keywords used for subsequent text queries

[4] http://www.chiariglione.org/mpeg/

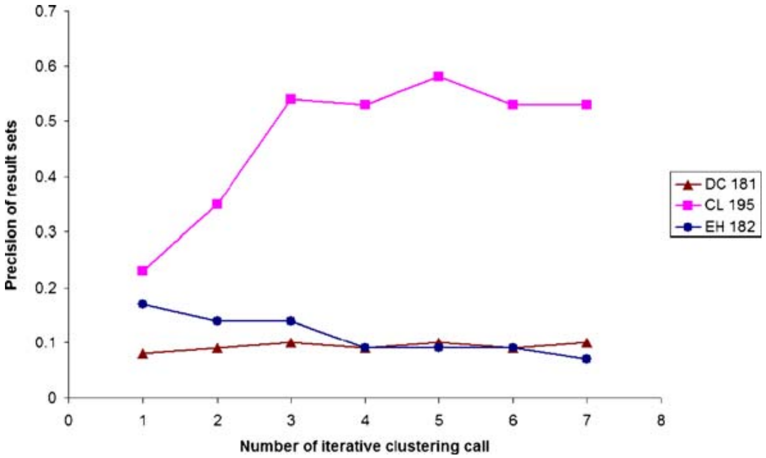


Fig. 8 Examples of different precision evolutions after few iterative clustering calls

are selected on the basis of their frequency in the documents of each cluster. We assume that the more frequent the keyword is, the more pertinent it is for the cluster. The results of the iterative clustering algorithm on this text query expansion experiment are illustrated in Table 2. It can be seen that using text query expansion in the iterative clustering algorithm is a good approach to improve the precision of the retrieval even if for most of the topics, the precision is “stable”.

Figure 9 presents typical examples of the evolution of precision for our experiments with one keyword added. The graph illustrates that adding one keyword after another to the previous query will slowly change the precision value.

These results on users’ textual or visual queries demonstrate that using the iterative clustering algorithm can improve the retrieved results for selected tasks. Most of the topics have a stable precision after few iterative calls of our clustering algorithm using visual or text features. However, this “stable” result is not useless, it means that the iterative clustering algorithm retrieves as many relevant documents as the initial text query. For some topics, these new results contain new relevant documents which have not been retrieved before. Thus, new documents are retrieved via a new facet in the faceted browser. We focus on this aspect in the following experiments.

4.4 Selecting performing topics

These results on text queries show that most of the topics have a “stable” precision, that means our iterative algorithm has little effect on the new facets compared to the original one. This is often due to the poor precision of initial results. If we look closer

Table 2 Results using text query expansion

Text query expansion	−	=	+
Add 1 keyword	2	19	3
Add 2 keywords	5	15	4
Add 3 keywords	5	16	3

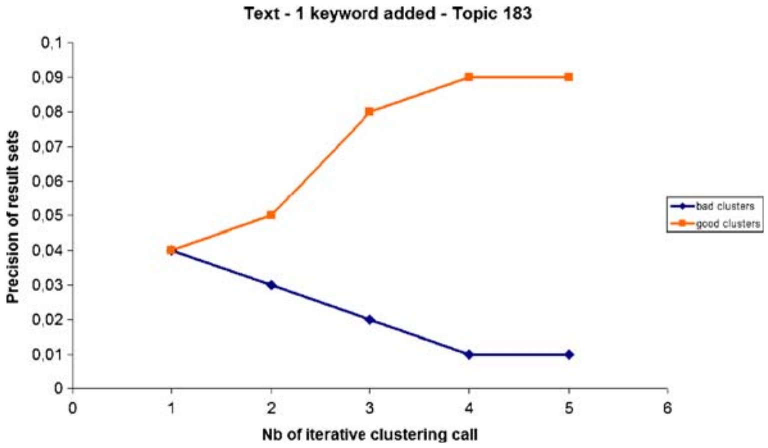


Fig. 9 Tendency of precision between bad and good clusters—one keyword added to text query

at the initial precision of these topics, we observe that 11 of the 24 topics have an initial precision of under 5%. Such a low precision will affect the pertinence of the keywords chosen for the query expansion. Hence, in the next set of experiments, we select specific performing topics and base our results on only the 13 topics that have a initial precision of retrieved results higher than 5%.

Experiment on visual queries In Table 3, we present the results of these tasks using visual queries. Column 3 shows the number of topics where the best cluster achieved a precision above 5% for each of the features. The last two columns present the number of topics for which visual feature iterative clustering has either no effect or a positive effect. A positive effect means that our approach successfully presents new interesting facets, i.e. facets with higher precision values than the original query or facets with at least 5% precision coming from different relevant documents than the initial text query.

For example, our iterative clustering algorithm gives a precision value higher than 5% for 7 topics for the visual feature "texture" and for 9 topics, our approach has a positive effect on the retrieved results which shows that using this visual feature to create a new facet is promising. Hence, new relevant results will be displayed in new facets.

We observe that most of the "stable" results in Table 1 are part of the "positive effect" column of Table 3, so they retrieve new relevant results which have not been

Table 3 Results using only visual features queries

Visual features	Prec. <5%	Prec. ≥ 5%	No effect	Positive effect
Dominant colour	11	2	5	8
Colour layout	9	4	7	6
Texture	6	7	4	9
Edge histogram	3	10	3	10
Contour shape	13	0	9	4

retrieved before. Consequently, incorporating different facets, the interface displays more relevant documents. This indicates that using iterative clustering on visual features can improve the precision of the results.

The results in Table 3 support the previous observation: edge histogram and texture features improve the effectiveness of the faceted browser the most. They also confirm that the contour shape visual feature is not working well with this database.

Experiment on text queries Table 4 shows the result of our iterative clustering approach on query expansion using one, two or three terms to expand the query. We observe here that an expansion using two keywords is enough to ameliorate the effectiveness of the retrieved results. Those results show that for six out of the 13 topics, the introduced approach returns a total of 63 new relevant documents, i.e. an average of 10 new relevant documents per topic compared to the initial query. However, we observe that for the other 7 topics, the precision of retrieved results stays "stable" which means that we require a different retrieval model to increase the performance of faceted browsing.

4.5 Focusing search on facets

We conduct a set of experiments to evaluate the new retrieval approach. The objective is to reduce the redundancy between facets. Based on an initial text query, we cluster the results and retrieve a list of keywords for each cluster. We then propose new facets based on new text queries. These text queries contain k new keywords. However, keywords of one facet's text query will not be used in another facet's text query. We simulate actions of users selecting the most relevant facets and launch an iterative call of our clustering algorithm on the new results, as used above.

The new retrieval model has a strong impact on the retrieval performance. Even after a few iterative calls, we retrieve few queries that seem similar in term of retrieved results compared to the original query. Figure 10 shows the difference between the precision of two facets using 5 new keywords. Table 5 presents the results we obtain for different values of k. We choose to evaluate this retrieval model for $2 \leq k \leq 6$, denoted "new 2" to "new 6" respectively. We observe better results with the "4 new keywords" than with other values of k, increasing the precision of 11 topics out of 13 or 15 out of the 24 initial topics. Using "4 new keywords" generates the highest number of new relevant documents compared to other values of k. This shows the effectiveness of this retrieval model compared to the text expanded query. It also shows that a combination of different retrieval models could help to improve the effectiveness of the faceted browser.

Table 4 Results using text query expansion

Text Queries	No effect	Positive effect	Number of new relevant documents
Add 1	6	7	40
Add 2	4	9	63
Add 3	5	8	51

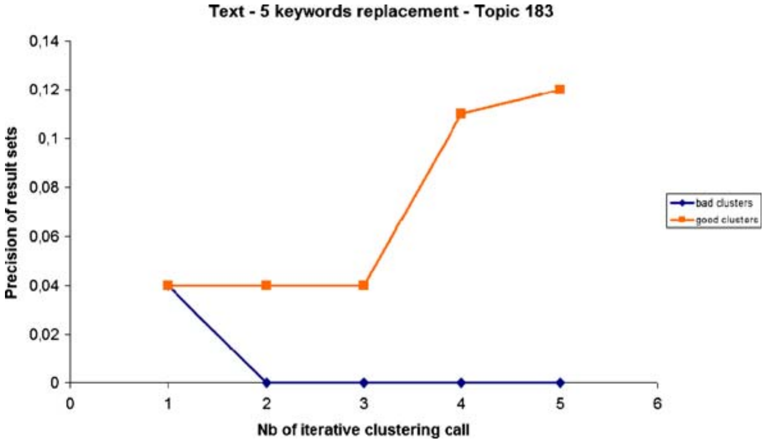


Fig. 10 Tendency of precision between bad and good clusters—5 new keywords as text query

4.6 Combined simulation with all features

In this section, we consider the best facets obtained by individual features. The idea here is not to combine all features in one query but to present every feature in different facets, so the user can choose the relevant features and have a faceted browser showing many more relevant documents than the initial retrieved results. In the presented results, we use the best text retrieval model based on the previous results: a query expansion of two keywords and the "4 new keywords" model.

Figure 11 shows the evolution of the number of relevant documents displayed in the faceted browser with respect to the number of facets/features used. We observe that the more facets/features we combine, the more relevant documents are retrieved. Figure 12 is a zoom of the box in Fig. 11 and focuses on the most relevant combinations of three facets which are texture, edge histogram and one of the text feature, query expansion or new 4 keywords, and the less relevant combinations of five facets which contains both dominant colour, contour shape and both text features. These results show that the texture and edge histogram seems to be the best visual features to combine and also that using only one of the two text query models is enough.

Finally, we present in Table 6 the best combination of features to obtain the best relevance for the faceted browser. A "–" means that we do not use the feature in the combination and an "×" means that the feature is part of the combination. Each column represents a feature. We denote "DC", "CL", "T", "EH", "CS", "TxA"

Table 5 Results using *k* new keywords as text queries

Facets with text queries	No effect	Positive effect	Number of new relevant documents
New 2	3	10	40
New 3	3	10	51
New 4	2	11	57
New 5	4	9	49
New 6	3	10	45

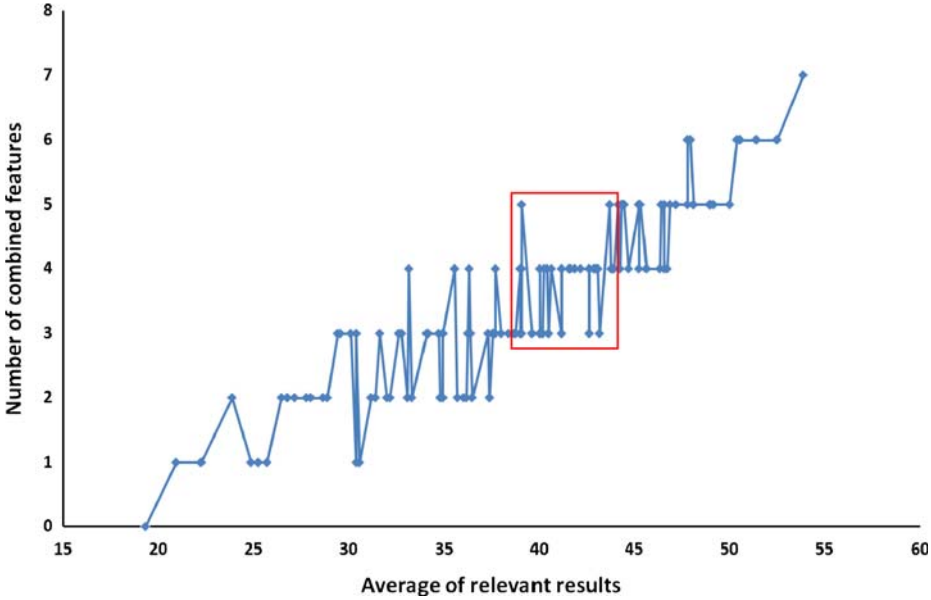


Fig. 11 Average number of relevant documents displayed in the interface with respect to the number of combined features used

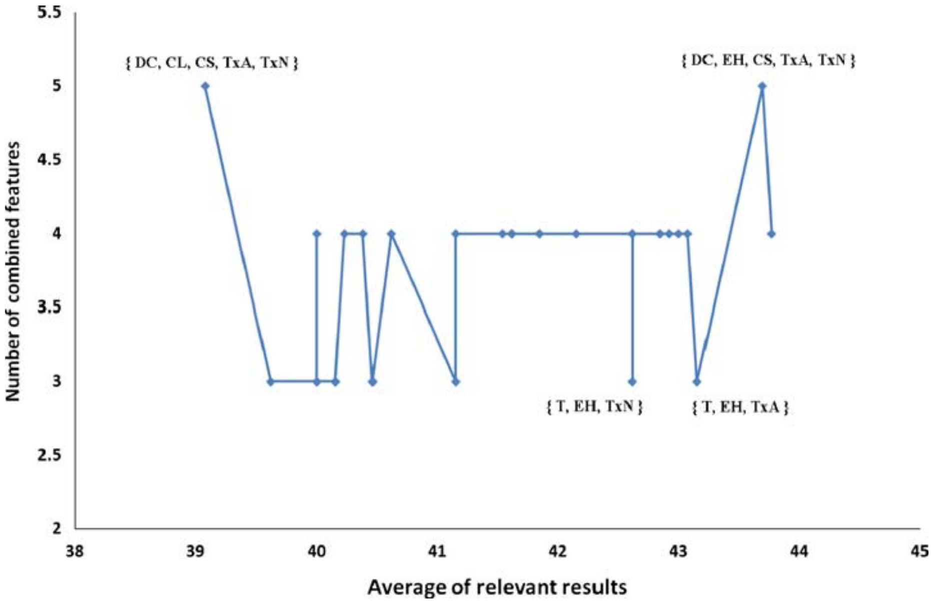


Fig. 12 Average number of relevant documents displayed in the interface with respect to the number of combined features used—zoom

Table 6 Best combinations of features

DC	CL	T	EH	CS	TxA	TxN	ARD
×	×	×	×	×	×	×	19.3
×	–	×	–	×	–	×	43.1
	×	–	–	–	–	–	47.8
–	–	–	×	–	–	–	47.9
×	–	–	–	–	–	×	48.1
×	–	×	–	–	–	–	48.1
–	–	–	–	×	×	–	48.9
–	–	×	–	×	–	–	49
–	–	–	–	×	–	×	49.1
×	–	–	–	×	–	–	50
–	–	–	–	–	×	–	50.4
–	–	–	–	–	–	×	50.5
–	–	×	–	–	–	–	50.5
×	–	–	–	–	–	–	51.4
–	–	–	–	×	–	–	52.5
–	–	–	–	–	–	–	53.8

and “TxN” for dominant colour, colour layout, texture, edge histogram, contour shape, text query expansion adding 2 keywords and text query with 4 new keywords, respectively. The last column shows the average number of relevant documents per topic denoted “ARD”. The first row shows the baseline run with no combination of facets, the second row presents the best combination of three feature, the next rows show the top combination of features. Thus, the last row shows the results of all feature combinations. Observing these results, we conclude:

- Colour layout, texture and edge histogram are the best visual features as they are almost always used in the top combination of features;
- Contour shape visual feature is almost useless as we improve the average number of new relevant documents per topic by only one in the combination (see the difference between the two last rows of the table);
- The effectiveness of the faceted browser can be more than doubled using a combination of two or more features. For example, the initial text query which has an average of 19.3 relevant documents per topic and the best three combination of features has an $ARD = 43.1$ or with all combined features an $ARD = 53.8$.

User Experiment In this section, we present a user study which aims to investigate the user view of the usefulness of low-level visual features. 12 participants took part in our evaluation. The participants were mostly postgraduate students and researchers at university, and indicated that they regularly interacted with and searched for multimedia. The experiment took approximately half an hour. Users were asked to mark some of the retrieved documents from 10 different query keyframes as “relevant” or “irrelevant”. The query keyframe was randomly chosen by the interface from the topics of the TRECVid 2006 collection and best 20 retrieved results using each visual feature were displayed randomly on the result interface. Table 7 presents the results of this user experiment. A total of 120 query keyframes were used representing 2400 retrieved keyframes for each low-level visual feature.

Table 7 User judgement on relevant documents based on low level visual features

	DC	CL	T	EH	CS
Marked as relevant	446	683	692	745	80
Marked as irrelevant	832	856	753	730	1103
Non marked	1122	861	955	925	1217

We noticed that more than half of the retrieved images were marked by users. This study confirms our previous results:

– Contour shape visual feature is not useful based on the TRECVid corpus. Indeed, less than 4% of the retrieved documents were marked as relevant and half of them were marked as irrelevant.
– Edge histogram, texture and color layout help to retrieve more than 28% of relevant documents which represents a high value compared to the precision results obtained before. This result can be explained by the fact that keyframes selected to represent the content of the video shots are not representative. For example, a keyframe showing a news caster is visually relevant when compared to another news caster keyframe, however the topic presented is probably different.
– The dominant color feature presents very good results only for few topics such as "sport", suggesting that it might be useful.

Discussion In this section, we have presented various experiments which aim to show the potential benefits of the faceted browser by modelling the user behaviour. It has been demonstrated that for most of the topics, visual or textual features can work using the iterative clustering methodology. Our results highlight the fact that new facets created by iterative calls of the clustering algorithm can increase the precision of the retrieved results and have a higher probability of displaying new relevant documents in new facets of the interface. This methodology shows potential to narrow the existing semantic gap problem.

One issue we encountered is that the poor textual annotation of the TRECVid corpus limits the effectiveness of the initial search queries and consequently the results of our iterative clustering approach. Thus, we had to focus on selected tasks for our experiments.

We have also presented a new text retrieval model, creating various new text queries, that is specifically designed to retrieve new relevant documents rather than to refine the precision of an initial text query. The results have shown the effectiveness of such a retrieval model: more topics received a positive effect than with the query expansion retrieval model. We have evaluated all possible combinations of the best simulated facets representing one feature each which shows the real potential of the faceted browser. We observed a real benefit combining facets with new results. Indeed, the number of relevant documents displayed doubles for a combination of three facets and almost triples with all facets.

Our fundamental premise in our simulated study is that users act to maximise the retrieval of relevant documents. For example, in an interactive user scenario, we assume that users choose better relevant clusters or keywords to add to a new facet. He or she may also easily delete a facet that does not correspond to their search task, which we presume will result in much better results with real user interactions than with our simulated clustering methodology.

5 Exploiting log files

In order to verify the above results, we conducted another set of simulated experiments based on logged data from a user experiment on the system described in Section 2.2. The user study studied the user perception, satisfaction and performance of the faceted browser, a brief overview being provided in Section 5.1. Exploiting the log files recorded from this study, we introduce and evaluate a new retrieval model which updates search queries by incorporating the content of other facets. The approach will be introduced in Section 5.2.

5.1 User experiment

In the user experiment [28], two tasks were defined, aiming to reflect two separate broad user needs. Task A was the more open of the two tasks, and asked the user to discover material reflecting international politics at the end of 2005 (the period of time covered by the TRECVid 2006 data). Task B asked for a summary of the trial of Saddam Hussein to be constructed, including the different events which took place and the different people involved (such as the judge). This later task, which is still multi-faceted, was less open ended than the former task. 24 subjects took part in the study. 12 users performed search Task A and 12 participants performed search Task B for 30 minutes and filled in a questionnaire.

5.2 Methodology

5.2.1 Identifying usage patterns

After performing the initial user study, we analysed the resulting log files and extracted user behaviour information. The following data was captured in the logs:

- *Creating a new facet*: Creating a new facet.
- *Deleting a facet*: Removing an existing facet.
- *Search*: Triggering a new retrieval in facet
- *Moving from facet*: Moving a shot from the relevance list of facet F_1 to a different facet F_2.
- *Dragging from player*: Dragging a shot from the video player directly onto a relevant results list of a facet.
- *Dragging from results*: Dragging a shot from a results list onto a relevance list.

The log entries provide us with information about the users' interaction behaviour such as when a user created a new facet, which search query he/she triggered or which results he/she judged to be relevant for this particular facet. We exploited this information in our simulation.

Figure 13 shows an example search session where a user interacts for 30 minutes with the facet browser. Note that this is a simplified graphic that does not contain usage information such as moving shots to a relevant result list. Within this session, the user first triggers two searches in facet F_1, creates a second facet F_2 and triggers

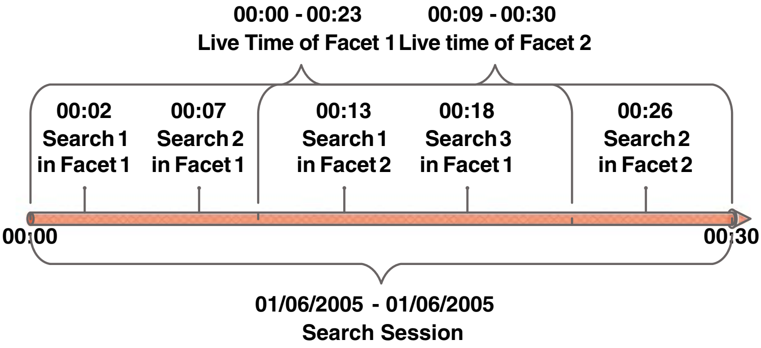


Fig. 13 An example user session

a search in it. Afterwards, he triggers a search in F_1, closes F_1 and executes another search in F_2. Let us define the event of triggering a search in a facet as the beginning of a *search iteration* in the facet and the triggering of a new search or of closing a facet as the *end* of a search iteration. In Fig. 13, we then have three iterations in F_1 and two iterations in F_2. Let us further define the beginning of a search iteration in any facet as the beginning of a new *search step* in the whole search session. In the example session shown in Fig. 13, the first step starts after 00:02 minutes with F_1 being in the first iteration. Step two starts after 00:07 minutes with F_1 being in the second iteration. With the start of step three after 00:13 minutes, F_1 is still in iteration two and F_2 starting the first iteration.

In the following, we use these patterns to study how facet based browsing can influence the retrieval performance in repeating users' interaction steps and updating the retrieval results.

5.2.2 Relevance judgements

Since Tasks A and B are not from TRECVid, ground truth data for our simulation was based on pooling [24] all sets R_i of shots d moved to the relevance list by user i. Let $\vec{d}_K =$ be a vector representing shot K, defined as

$$\vec{d}_K = \{d_{K1} ... d_{KN}\}, \text{ where } N \text{ is the number of users} \tag{3}$$

and

$$d_{Ki} = \begin{cases} 1, & \vec{d}_K \in R_i \\ 0, & \text{otherwise} \end{cases}$$

Using:

$$F_1\left(\vec{d}_K\right) = \begin{cases} 1, & \left(\sum_{i=1}^{N} \vec{d}_{Ki}\right) = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$F_2\left(\vec{d}_K\right) = \begin{cases} 1, & \left(\sum_{i=1}^{N} \vec{d}_{Ki}\right) \geq 2 \\ 0, & \text{otherwise} \end{cases}$$

we created two relevance judgement lists:

$$L_1 = \left\{ d_K : F_1\left(\vec{d}_K\right) = 1 \right\} \tag{4}$$

(Assuming that a keyframe is relevant within the given topic when it was selected by any user.)

$$L_2 = \left\{ d_K : F_2\left(\vec{d}_K\right) = 1 \right\} \tag{5}$$

(Assuming that a keyframe is relevant within the given topic when it was selected by at least two users.)

5.2.3 Simulation strategies

The user retrieval model for our study was simple: users enter textual search queries in each facet and the backend system returns a list of shots which are represented by a keyframe in the result list of the facet. We simulate users interacting with the result list by selecting relevant shots, playing a shot, creating facets, etc. However, user feedback such as selecting a shot as relevant for this facet or the content and status of other facets are not used in retrieving or suggesting new facets. Hence, we use the user study as a baseline run B and try to improve its retrieval performance by introducing a new retrieval model which incorporates the content of other facets.

Our simulation procedure is as follows. First of all, we analysed the user queries in the log files and confirmed that users took advantage of the facets and used them to search for variations of the same concept. For instance, in Task A (international politics), participants used the facets to search for different politicians, i.e. “George Bush” in facet F_1 and “Tony Blair” in facet F_2. In Task B (trial of Saddam Hussein), facets were used to search for different events during the trial from “Saddam’s capture” to “his execution”. We concluded that facets were used to focus more on specific sub concepts of each topic. Following this, we performed a simulation run S.

In this run, we took advantage of the explicit relevance feedback given by each user in marking shots as relevant for a facet. We used these shots as a query expansion source and determined query candidate terms for each iteration in each facet by expanding queries from the relevant rated keyframes at step x. If a term appears in more than one facet within this step x, we removed it from the facet which contained more candidate terms and used these candidate terms as a new search query. In other words, we reduce the number of query terms in a facet, when the query term is used in another facet with less query terms at the same time. This results in a more focused retrieval for the facets, as double entries will be avoided.

Table 8 Example association of steps and facet’s iterations

#Step (time)	Facet (iteration)
1 (00:02)	1(1)
2 (00:07)	1(2)
3 (00:13)	1(2) and 2(1)
4 (00:18)	1(3) and 2(1)
5 (00:26)	2(2)

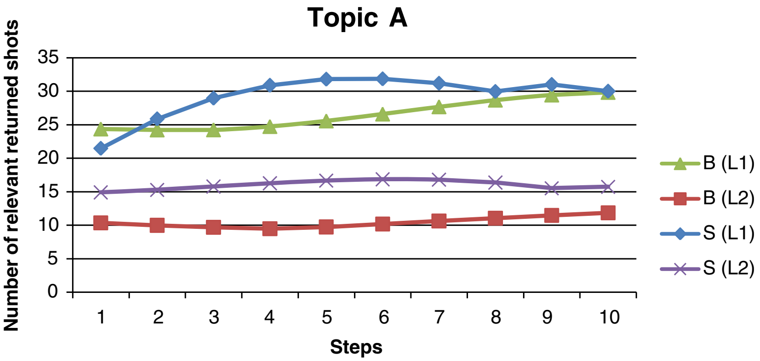


Fig. 14 Number of relevant returned results over all steps in Topic A

5.3 Results

To evaluate the performance of our baseline system and the simulation run, we firstly divided the users's search sessions into separate steps, being the beginning of a new iteration in any facet. For each step, we then combined the result lists of each facet in its current iteration. Table 8 presents the steps and facet iterations that can be identified using the example session shown in Fig. 13.

In the next step, we evaluated our runs using the two created relevance judgement lists L_1 and L_2 as introduced in Section 5.2.2. Figures 14 and 15 show the mean number of relevant retrieved results over all steps in Topic A and B, respectively. As expected, using the relevance judgements list L_1 gives a higher retrieval performance in all cases than using L_2. This matches common sense, a larger list of relevant documents used for evaluation results in a higher number of relevant retrieved documents. The decreasing number of retrieved shots in some cases is the direct consequence of users closing facets in later steps of their retrieval session. The results within these facets hence get lost, resulting in a decrease of retrieved results.

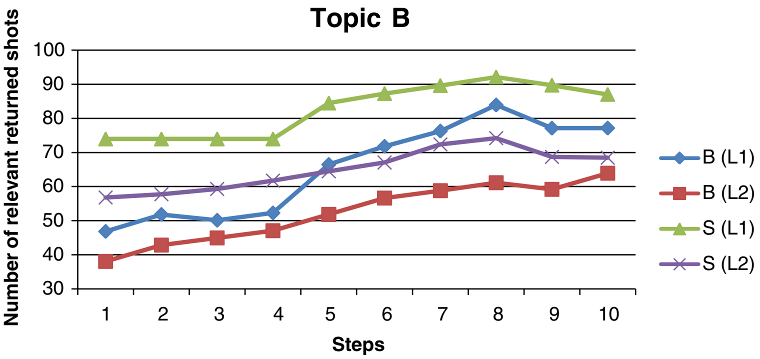


Fig. 15 Number of relevant returned results over all steps in Topic B

Discussion It can be seen that for both search tasks, the simulation run S outperformed the baseline run B, which indicates that considering the content of other facets to focus a user's search query can improve the retrieval performance. Hence, a retrieval model which takes the content of other facets into account can outperform a classical "one result list only" model. This conclusion supports the outcome of our simulation of the user behaviour presented in Section 4.5: a retrieval model adapted to a facet-based system has the potential to enhance retrieval effectiveness.

6 Conclusion

In this paper, we have evaluated a facet-based approach to interactive video retrieval. Such an approach has the potential to address the semantic gap issue, by allowing users to explore their interest in various aspects of a task. However, as it uses the low level visual features, this approach faces the "curse of dimensionality". We have presented the potential performance bounds of such a system.

First, we have proposed a methodology to select the most appropriate dimensions of each feature. Our experiment has shown the potential of feature selection and dimensionality reduction. This method can be useful to overcome the "curse of dimensionality" with a minimum loss in precision. Such a method potentially allows faster querying on different low level features. Interactive retrieval systems can benefit from this gain of speed, especially systems supporting facet-based browsing, where multiple searches may be carried out at the same time.

Due to the lack of an appropriate evaluation methodology for facet-based retrieval, we have proposed a simulated evaluation methodology which models user interactions. We have described such a scheme which employed clustering to identify potential facets created by users. This methodology uses both textual and visual features.

The results of our study demonstrate the potential benefits of a faceted search and browsing system. It is clear from the study that there are tasks which benefit from such an approach. In addition to the results of our simulated evaluation on the TRECVid collection, we have explored the logs of a real user-centred evaluation and the results corroborate that of the simulation methodology. We have also explored the possibility of enhancing retrieval performance by the use of appropriate retrieval models. Clearly, the results show the benefits and also the possibility of employing more advanced models.

The experiments are conducted on a large data set (TRECVid 2006) and hence support the validity of our experiments. However, it is well known that the TRECVid search topics are diverse and there is the issue of the performance variation between topics. This may explain some of the performance problems we encountered in some of the topics. In addition to this, simulated methodologies are at one end of spectrum of a series of evaluations ideally required before multimedia systems are deployed. It allows us to benchmark various retrieval approaches and search strategies such as faceted browsing. However, it is important to verify the results of simulations via the use of a user-centred evaluation, which is being explored at the moment.

Acknowledgements This research was supported by the European Commission under contract FP6-027122-SALERO. The third author was supported by the JCCM under project (PCI08-0048-

8577), MEC under project (TIN2007-67418-C03-01) and FEDER funds. It is the view of the authors but not necessarily the view of the community.

References

1. Bekkerman R, McCallum A, Huang G (2005) Automatic categorization of email into folders: bechmark experiments on enron and sri corpora. Technical report, Department of Computer Science. Amherst, University of Massachusetts
2. Bermejo P, Gámez J, Puerta J, Uribe R (2008) Improving knn-based e-mail classification into folders generating class-balanced datasets. In: IPMU'08: proceedings of the 12th intl. conf. on information processing and management of uncertainty in knowledge-based systems
3. Finin TW (1989) GUMS: a general user modeling shell. User models in dialog systems, pp 411–430
4. Flores MJ, Gámez JA, Mateo JL (2007) Mining the esrom: a study of breeding value classification in manchego sheep by means of attribute selection and construction. Comput Electron Agric 60(2):167–177
5. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305
6. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
7. Halvey MJ, Keane MT (2007) Analysis of online video search and sharing. In: HT '07: proceedings of the eighteenth conference on hypertext and hypermedia. ACM, New York, pp 217–226
8. Harper DJ, Kelly D (2006) Contextual relevance feedback. In: IIiX: proceedings of the 1st international conference on information interaction in context. ACM, New York, pp 129–137
9. Hersh W, Over P (2000) TREC-8 interactive track report. In: The eighth text retrieval conference (TREC 8)
10. Hopfgartner F, Jose J (2007) Evaluating the implicit feedback models for adaptive video retrieval. In: ACM MIR '07, September, pp 323–332
11. Hu Y-J (1998) Constructive induction: covering attribute spectrum. In: Feature extraction, construction and selection: a data mining perspective. Kluwer, Dordrecht
12. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall, Englewood Cliffs
13. Kerne A, Koh E, Smith S, Choi H, Graeber R, Webb A (2007) Promoting emergence in information discovery by representing collections with composition. In: C&C '07: proceedings of the 6th ACM SIGCHI conference on creativity & cognition. ACM, New York, pp 117–126
14. Larsen O, Freitas A, Nievola J (2002) Constructing x-of-n attributes with a genetic algorithm. In: Proc genetic and evolutionary computation conf (GECCO-2002)
15. Liu H, Motoda H (1998) Feature extraction construction and selection: a data mining perspective. Kluwer, Dordrecht
16. Nakazato N, Manola L, Huang TS (2002) Extending image retrieval with group-oriented interface. In: Proceedings of advanced visual interfaces
17. Over P (1999) TREC-5 interactive track report. In: The seventh text retrieval conference (TREC 5)
18. Quinlan J (1986) Induction of decision trees. Mach Learn 1:81–106
19. Robertson SE, Walker S, Jones S, Hancock-Beaulieu M, Gatford M (1994) Okapi at TREC-3. In: Proceedings of the third text retrieval conference (TREC 1994), Gaithersburg
20. Rudinac S, Zajic G, Uscumlic M, Rudinac M, Reljin B (2007) Comparison of cbir systems with different number of feature vector components. In: SMAP '07: proceedings of the second international workshop on semantic media adaptation and personalization. IEEE Computer Society, Washington, DC, pp 199–204
21. Shen HT, Ooi BC, Zhou X (2005) Towards effective indexing for very large video sequence database. In: Proceedings of the ACM SIGMOD international conference on management of data. ACM, Baltimore, pp 730–741
22. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and trecvid. In: MIR '06: proceedings of the 8th ACM international workshop on multimedia information retrieval. ACM, New York, pp 321–330
23. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380

24. Spärck-Jones K, van Rijsbergen CJ (1975) Report on the need for and provision of an ideal information retrieval test collection. Technical report, University Computer Laboratory, Cambridge, British Library Research and Development report 5266
25. Urban J, Jose JM (2004) Ego: a personalised multimedia management tool. In: Proc. of the 2nd int. workshop on adaptive multimedia retrieval, pp 3–17
26. Urruty T, Djeraba C, Jose JM (2008) An efficient indexing structure for multimedia data. In: MIR08: international conference on multimedia information retrieval. ACM, New York, pp 313–320
27. van Rijsbergen CJ (1979) Information retrieval, 2nd edn. Butterworths, London
28. Villa R, Gildea N, Jose JM (2008) A faceted search interface for multimedia retrieval. In: SIGIR'08, Singapore, pp 775–776
29. Villa R, Gildea N, Jose JM (2008) FacetBrowser: a user interface for complex search tasks. In: MM'08: international conference on multimedia, Vancouver, pp 489–498
30. Webb G, Boughton J, Wang Z (2005) Not so naive bayes: aggregating one-dependence estimators. Mach Learn 58(1):5–24
31. Weber R, Schek H-J, Blott S (1998) A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: Proceedings of 24rd international conference on very large data bases. Morgan Kaufmann, New York, pp 194–205
32. White AP, Liu WZ (1994) Technical note: bias in information-based measures in decision tree induction. Mach Learn 15(3):321–329
33. White R, Bilenko M, Cucerzan S (2007) Studying the use of popular destinations to enhance web search interaction. In: ACM SIGIR '07—proceedings of the 30th international ACM SIGIR conference, Amsterdam, July, pp 159–166

Frank Hopfgartner is a doctoral candidate in information retrieval at the University of Glasgow, Scotland. He received a Diplom-Informatik degree from the University of Koblenz-Landau, Germany in 2006. His research interests include interactive video retrieval with a main focus on relevance feedback and adaptive search systems. Frank is a member of BCS, BCS IRSG and ACM SIGIR.

Thierry Urruty is a post doctoral researcher at the University of Lille, France. He finished his PhD end of 2007 in multimedia indexing. In 2008, he has been a research assistant in the Information Retrieval Group at the University of Glasgow. His research interests are data mining, video indexing and retrieval and more recently computer vision.

Pablo Bermejo Lopez is a PhD Student in the Intelligent Systems and Data Mining group (SIMD) in Castilla-La Mancha University (UCLM), Spain. He received his degree in Computer Science Engineering in 2004 and finished his Master Thesis in 2007, both degrees also in UCLM. His research is mainly focused in pre-processing of high-dimensional databases for supervised classification, including feature selection, feature construction, instances balancing and context influence.

Robert Villa holds a PhD in Computing Science from the University of Glasgow. After working at the Consiglio Nazionale delle Ricerche in Milan, Italy and at the University of Strathclyde, UK, he is now with the Information Retrieval Group at the University of Glasgow. He is mostly focusing on the user aspects of content-based video retrieval systems.

Joemon M. Jose is a Professor at the Department of Computing Science, University of Glasgow. His research focuses around the following three themes: (i) Adaptive and personalized search systems; (ii) Multimodal interaction for information retrieval; (iii) Multimedia mining and search. He has published widely in these areas and leads the Multimedia Information Retrieval group (mir.dcs.gla.ac.uk). He has a PhD in information retrieval, MS in Software Systems and MSc in Statistics. He is a Fellow of BCS, member of the ACM, IEEE and IET professional societies.