Mutual Relevance Feedback for Multimodal Query Formulation in Video Retrieval

Arnon Amir
IBM Almaden Research Center
650 Harry Road, B2-408
San Jose, CA 95120
Phone: +1-408-927-1946
arbon@almaden.ibm.com

Marco Berg
IBM BCS - Media & Entertainment
Wilhelm-Fay-Strasse 30-34,
65936 Frankfurt, Germany
Phone: +49-69-6645-1349
mber1@gmx.de

Haim Permuter
Electrical Engineering Dept.
Packard Elect Eng Bldg
Stanford University
Stanford, CA 94305-9510
Phone: +1-650-724-3633
haim1@stanford.edu

ABSTRACT
Video indexing and retrieval systems allow users to find relevant video segments for a given information need. A multimodal video index may include speech indices, a text-from-screen (OCR) index, semantic visual concepts, content-based image features, audio features and more. Formulating an efficient multimodal query for a given information need is much less intuitive and more challenging for the user than of composing a text query in document search. This paper describes a video retrieval system that uses mutual relevance feedback for multimodal query formulation. Through an iterative search and browse session, the user provides relevance feedback on system’s output and the system provides the user a mutual feedback which leads to better query and better retrieval results. Official evaluation at the NIST TRECVID 2004 Search Task is provided for both Manual and Interactive search. It is shown that in the Manual task the queries result from the mutual feedback on the training data significantly improve the retrieval performances. A further improvement over the manual search is achieved in the interactive task by using both browsing and mutual feedback on the test set.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Relevance feedback, Search Process, Selection Process.

General Terms
Algorithms, Performance, Experimentation, Human Factors.

Keywords
Multimedia, Video Retrieval, Relevance Feedback, Multimodal Search, Query Formulation, Query Refinement, NIST TRECVID.

1. INTRODUCTION
In the era of information explosion, an enormous amount of documents of all types and origin are available at the tip of our hands. Search and browse methods take a more important role than ever before. As the amount of digital images, audio and video grows rapidly, both on the Internet and on user’s personal computers, efficient multimedia search and browse are becoming essential productivity tools.

Multimedia documents, in which information is implicitly represented by audio samples, pixel colors and motion vectors, are much harder to index and search than text documents. While text documents are encoded and searched using words in natural human language, there is yet no such broad consensus about document annotation and query representation when searching a collection of images, audio and video documents. Should a query be composed of words, should it contain sample images, sample audio or perhaps motion vectors? What query should one use to search for “people walking with umbrellas” in a video collection? Would you search for it by image samples, color, texture, shape, speech, semantic visual concepts, or any combination of the above? And what user interface would you expect to use in order to formulate such a query? This example is one of the search topics provided by the National Institute of Standards and Technology (NIST) at the TRECVID 2004 video search evaluation [16].

The work presented in this paper may be perceived as a step towards the transformation of a search topic – a definition of the information need, into a search query – the actual input to a search engine. It is widely regarded as one of the most challenging tasks in information retrieval, in text and even more so in multimedia. A topic description for multimedia search may include a textual description, images, sketches, audio and video samples [16]. The search query may be composed of the same modalities, a subset of them, and/or other modalities.

The main contribution of this paper is the cross-modal query expansion using mutual relevance feedback. A multimodal query is formulated using a single text box, allowing search in different search modalities, such as words in speech transcript and closed captions, text in video OCR, search for semantic visual concepts and referencing any relevant key-frames for Content Based Image Retrieval (CBIR). A user-centric, penetrable and highly interactive mutual relevance feedback assists the user along the query formulation process. Tightly coupled search and browse operations provide efficient navigation in a large collection of videos and rapid accumulation of relevant video segments without loosing the context of the search session. System’s respond to consequent iterations is complete and consistent with accumulated session knowledge. At the end of a search session, the user ends up with not only the search results but also a well-crafted multimodal search query that may be kept as a meta-representation of the search topic and may be
applied to the video collection at later times, or to other video collections, either by the same user or by others. Last, the important of efficient browsing is evaluated by quantifying the contribution of manual shots elevation in interactive search. It is shown that for optimal performance within a given interaction time frame, a balance between query formulation (search) and browsing has to be kept.

The rest of this paper is organized as follows. Section 2 provides a brief review of work on relevance feedback in text and in multimedia search. An overview of our multimodal indexing and search system is given in Section 3. Section 4 describes the mutual relevance feedback, including the user experience, iterative sessions with tightly coupled search and browse. Section 5 describes the evaluation of this system at the NIST TRECVID 2004 Search task. Section 6 concludes and summarizes the results of this study.

2. OVERVIEW: RELEVANCE FEEDBACK IN TEXT AND IN MULTIMEDIA

Relevance feedback (RF) in text retrieval has been studied for more than forty years (see, e.g., [22]) and is a much more mature field than its multimedia counterpart. It aims at exploring the exploitation of usually a small number of feedback documents, labeled by a user, to improve topic search results in the overall corpus. In one of the earliest RF studies, Rocchio used a vector representation of queries and relevant items for query expansion and modification, where word’s weight is increased by positive feedback and decreased by negative feedback documents containing it [21]. Later statistical methods add the inter-document word frequency into the weight computation [4]. Robertson’s wpq method [19] was shown to be effective for selecting query terms (words, phrases) and re-ranking them, and later also for query expansion using other words from the feedback set [20]. In wpq and its many variants, word frequencies in both positive and negative documents are used to determine the word’s weight.

RF systems may modify the search ranking metric, or query to document similarity measure. CBIR search makes a classic example for multimedia search (see, e.g., [6][27] for review of RF techniques in CBIR). In a “find me more like this” image search, the initial search query may be a single image. Followed positive samples may be used to modify the weights assigned with different images features, such as color, shape, texture and other features in image-to-image comparison. A Minkowski distance in [24], and a Mahalanobis distance [12], are modified and governed by feature’s variance in the RF positive set. These approaches are based on the assumption that relevant images make ellipse-like cluster around a query single point. This assumption may be relaxed by clustering the multiple feedback samples, or the corresponding data points in the high dimensional feature space, and using a rich model such as K-NN (K nearest Neighbors), 1-class kernel SVM, or GMM (Gaussian Mixture Model) to model the more complex structure. Several such methods are cited and discussed in [27].

In recent work, RF in CBIR has been addressed by machine learning techniques, with the goal of increasing the discrimination power of the query, and of doing so as quickly as possible, with the least number of feedback documents. Support Vector Machine (SVM) is used for RF in text documents with notable success (e.g., [13]). Active learning techniques go one step further and select the next set of retrieved documents such that their labeling by the user is most likely to maximize the improved discrimination of uncertain documents in the next iteration. However, in the case of an iterative search task, the user expects to see a list of mostly relevant documents. Presenting a yet-another-set of uncertain ones may quickly lead to user frustration. Xu et al. [26] propose a hybrid RF approach in which a set of both relevant and uncertain documents is presented, to keep the user engaged in the active learning process. This dilemma is denoted as the “greedy vs. cooperative user model” [27]. While the method used in our work does not incorporate machine learning, we do pay attention to the way consequent search results are presented to the user in the context of user’s RF from previous iterations. We call it the system’s mutual feedback (to the user) and the principle of session consistency.

RF-based query modification techniques may be classified by the amount of automation vs. user’s involvement in the query reformulation. Koenemann and Belkin [15] compared three cases, from an Opaque, automatic system that hides the revised query from the user, through a Transparent system that shows the reformulated query and the revised search results to the user, to a Penetrable system, that only makes suggestions but leaves the actual query reformulation task to the user. The study shows a significant advantage in performance to users of the Penetrable system over those of Opaque or Transparent systems (those in turn perform better than a Control system with no RF). In a nice analogy to an automatic camera, Bates [3] suggests that while automatic cameras require only little user training, semi-automatic and manual modes may produce better results in the hands of experienced users. Hence a Penetrable system seems as the most promising approach for a research prototype, and is the selected choice for the system presented in this paper.

Human aspects and Human-Computer-Interaction (HCI) are widely studied in the context of RF, including the design of appropriate graphical user interfaces (GUI), efficient utilization of the display space, information visualization techniques, personalization, and group feedback. Without a careful design of the user experience, an otherwise promising RF approach might be found ineffective for use. Hearst’s review of RF and various user interface factors in information retrieval [11] was found very helpful in the design of our system.

3. Multimodal Query Processing

3.1.1 Indexing and Searching Modalities

At video ingesting time, the video frames and audio content are analyzed using multiple video analytics tools. The videos are segmented into shots, and each shot is associated with one or more representative key-frames. The following indices were generated and used in our experimentation. The indices are ordered by the level of semantic information they convey, from low-level image representation, through OCR and ASR and up to closed captions.

- **CBIR image index**: each key-frame is represented by a low-level feature vector, including color, texture and location features [2]. In addition, any sample images and I-frames of sample videos provided with each search topic are similarly processed. CBIR retrieval is based on simple nearest neighbors search in this features space.
- **Phonetic speech index**: a 1st-best phonetic transcript of the voice content is generated and indexed using metaphones for efficient sound-like search. The retrieval uses a Bayesian edit distance to find sound-like matches [1].
• **Semantic concepts:** forty multimodal detectors of semantic concept were applied at shot level, using visual, audio and speech ASR, using SVM and Multinet [2] trained on the TRECVID 2003 development data set and its common annotation in MPEG-7 [16]. The concepts list include “sky”, “sport”, “crowd”, “studio setup”, “car”, “face”, “person”, “weather”, etc.

• **Video OCR:** image text was detected in video frames and was automatically converted from image to text using OCR. [10]. At retrieval time, an open ends, uniform weights edit distance is used to overcome the frequent errors, typical to OCR of very low resolution video text from noisy frames.

• **Speech ASR:** a speech transcription is generated using speech recognition [7], follow by stemming and stop words removal [2].

• **Closed captions:** closed captions are automatically time-aligned to the speech in the audio track to compensate for any misalignments that occur during CC generation process and due to the separate video digitization and CC extraction processes.

### 3.1.2 Multimodal Search and Soft Boolean Ranking

A multimodal query is composed using a different word suffix for each of the search modalities. For example, the query “sport~ AND basketball AND (N.B.A.# OR NCAA$) AND 04.38333” would invoke search for the semantic (trained) concept “sport”, the word “basketball” in speech ASR and CC, the acronym “NBA” in phonetic speech index, the text “NCAA” in Video OCR and the frames similar to key-frame #38333 of data set TRECVID04 in the CBIR index. Each single modality search returns a separate list of matches. Those lists are then merged and segmented into “documents”, or video segments. A match is represented by a tuple <video, time, confidence, terms> containing 1 or more query-matching terms, of various modalities. This process ensures that nearby multiple instances of matched query terms are grouped together into one significant segment. It also fuses the multimodal information into a single score.

The individual matches are aggregated using the Soft Boolean (SFB) ranking method [18]. This method combines traditional OKAPI-based statistical ranking with Boolean-like query expressions. The computation is performed in two steps.

First, the initial score of each match is modified based on its contribution to fulfillment of the SFB expression. For a binary operator, if both operands are found within a certain time proximity, then the score of both operands is modified by the expression. An OR operator does not impact the individual score of its arguments. Hence for the expression $S_i OR S_k$, we have $S^{OR}_{i+k}=S_i$ and $S^{OR}_{i'}=S_k$, where $S_i$, $S_{i'}$ are the scores of term $i$ before updating the score and after, respectively. Fulfilling an AND operator is more rewarding to both of its operands. The scores of the operands in the expression $S_i AND S_k$ are updated as follows:

$$S^{AND}_{i} = S^{AND}_{k} = \max(S_i, S_k) + \alpha e^{\frac{x^2}{2T}}$$

$$S^{AND}_{i} = S_i$$

where $\alpha$, $T$, and $T_{\max}$ are constants. This process is applied recursively up the evaluation tree of the Boolean-like query expression. Precedence is given to parenthesis, AND and OR, in this order.

The second step is time-aggregation of the individual scored terms. This step is carried out using a modified OKAPI process, similar to [1]. The $tf*idf$ (term frequency times inter-document frequency) computation requires a definition of a “document”. In [1], “documents” are arbitrarily defined as (half-overlapping) video segments corresponding to speech chunks of 100 or 200 spoken words. Due to the multimodal nature of our search on one hand, and our need to retrieve very short shots on the other hand (TRECVID evaluates the retrieval of single shots), we slightly modify the definition of a “document” and make it a dynamic one. We sort all the scored terms of all modalities along the video:time axis, and then group them into segments based on the time gaps between consecutive terms. We start with the first scored term and keep adding more scored terms to grow a “document” as long as the time gap to the next term is smaller than $T_{\max}$. Hence a document may contain one or more query-matching terms, of various modalities. This process ensures that nearby multiple instances of matched query terms are grouped together into one significant segment. It also fuses the multimodal information into a single score.

Since the “document” boundaries are determined at query processing time, the standard OKAPI $idf$ value, which is the count of documents containing the term, is replaced with $tf$ - the total number of term occurrences found in the collection. Most of the retrieved segments are very short, typically only a few seconds. Hence $tf=1$ for most documents, supporting this substitution. Like the original OKAPI, this slightly modified version still scores rare query terms higher than frequent ones and gives higher scores to documents with multiple term occurrences.

The two steps SFB scoring further gives higher scores to documents which better fulfill AND clauses than to those which partially fulfill the same AND clause/s or those which fulfill OR clause/s instead. Under this definition, an OR-only query expression, e.g., $S_i OR S_k OR S_l$, falls back to an OKAPI search for $S_i$, $S_k$, $S_l$. A query like $S_i$ AND $(S_k OR S_l)$ would score higher a document containing $S_i$ and $S_k$ than one containing $S_i$ and $S_l$, assuming that their $tf$-s are the same.

### 4. Query Modification with Mutual Relevance Feedback

Figure 1 illustrates the Relevance Feedback process. A user submits a query $Q_0$ based on his information need. The query is being processed and a ranked result list of matching video segments is rendered. The user may then mark a subset of these segments as “positive” (relevant), or “negative” (non-relevant) and submit these marks as his relevance feedback. Based on the RF, the system comes up with suggested query terms to include in the query to create $Q_{i+1}$. The new query is then submitted for search. This process may proceed for several iterations, where the set of shots marked by the user as RF is accumulated along the session. Any new RF shots are being added to the RF pool of previous iterations.

![Figure 1: Iterative query formulation using relevance feedback.](image-url)
The goal of this iterative RF process is not just to improve the search results, but to help the user to formulate a better multimodal query. We compose the new result list from the top matches to the search and all the RF that was accumulated through the session, and use a colors code to mark these result lists with respect to previous ones. This novel way of composing the results list assists the user in judging the quality of his query compared to previous queries.

Hence we observe a duality in this iterative process, between the user and the search system. The user gives the system feedback about its search results, and the system gives the user feedback about his query quality. We call this duality mutual feedback, as feedback flows both ways. The rest of this section describes each of these feedbacks in detail.

**Term Ranking and Query Modification**

The user feedback is a set of shots, each marked as a positive or a negative shot. The system retrieves all the terms in these shots, ranks them using the feedback, and provides the user with a ranked list of suggested query terms. The score (weight), \( W_q \) of term \( i \) with respect to iteration (query and RF) \( j \) is calculated by the log likelihood ratio for this term to be present in relevant vs. non relevant video segments by

\[
W_q = \left( C \log \frac{ p_i \left( 1 - q_i \right) }{ 1 - p_i q_i } \right)
\]

where \( p_i \) and \( q_i \) are the estimated probabilities for term \( i \) to be found within the set of relevant and non relevant video segments, respectively. These probabilities are computed by

\[
p_i = \begin{cases} 
\frac{ r_i + 0.5 }{ R_i + 1.0 } & \text{when } r_i > 0 \\
0.01 & \text{when } r_i = 0 
\end{cases}
\]

\[
q_i = \frac{ n_i - r_i + 0.5 }{ N - R_i + 1.0 }
\]

where \( r_i \) is the number of occurrences of term \( i \) in the RF set, \( R_i \) is the total number of terms in the RF set, \( n_i \) denotes the number of occurrences of term \( i \) in the entire collection, and \( N \) is the total number of terms in the collection. Last, the constant \( C \) is selected for each modality, to allow fusion between search terms from speech, concepts, closed captions etc. We set \( C=0.2 \) for speech and \( C=1.4 \) for concepts based on experimenting in multiple runs with different values. Note that we also keep a separate term count \( N \) for each search modality (e.g. speech, visual concepts).

Negative feedback is similarly computed from negative RF documents. The purpose of negative feedback is to find good negative terms, which can be added to the query using the AND NOT operator. However, positive feedback is more likely to move a query closer to a user’s information need than negative feedback to push the query away from all the irrelevant shots. Positive query terms are more intuitive to use and have a more predictable effect on the search result [14], thus negative feedback was shown useful in pseudo RF [27]. Negative query terms may help, for example, in a case of concept ambiguity. If we go back to the basketball example, then “sport– AND basketball AND NOT hockey” may help to distinct basketball shots from other sport shots, like hokey, that are also marked by the semantic concept “sport”. But in some cases, negative query terms might reduce the score for relevant shots and reduce the query effectiveness. For example, when basketball shots are immediately followed in a video by hokey shots, that combined video segment will score less because of the negative terms it contains.

### 4.1.1 Graphical User Interface

A user interface for information retrieval should provide not only access to information by search and browse means but also assist users to reassess their intermediate results and adjust their search strategy. It should support search strategies by making it easy to follow trails with sometimes unanticipated results. This can be accomplished in part by supplying ways to record the progress of the current strategy and to store, find, and reload intermediate results, and by supporting pursuit of multiple strategies simultaneously [11]. Experiments in multimedia search demonstrated that interactive runs gain best performance with systems which allow more effective browsing and visualization of the search results of text queries using a variety of strategies, such as interactive storyboards, optimized color- and texture based image search engines or simplified classifier filter access [10].

Figure 2 shows the Graphical User Interface (GUI) of our system. It is built of a multimodal query formulation region (upper left), a Results List region (bottom, scrolls down) and an enhanced video streaming browser with multiple synchronized views (upper right).

![Figure 2: The GUI is divided into query formulation area (upper-left), video browsing area (upper-right) and results list (bottom). The new search terms suggested by the system are displayed in the query area. In the revised query results, shots with positive RF show in green (result 1 and part of 3) and shots with negative feedback show in red (result 2). All other shots keep their neutral background color.](image-url)
4.2 User-Centric Search Session

4.2.1 Interactive query formulation process
The main goal of formulating an effective multimodal query is pursued by both the user and the system in an interleaved interactive process (Figure 3). The two main circles in this diagram correspond to the query formulation process, supported by RF, and the result evaluation process, supported by browsing and mutual feedback. After sending a formulated query to the retrieval system, the results are rendered in a results list.

Figure 3: The user-centric interactive process. Search and browse operations are coupled with the relevance feedback and query modification process. Bi-directional arrows allow state transition in either way.

Figure 4: Browsing examples: a CBIR-board (left) and a storyboard (right). Each shot may be added to the relevance feedback pool using its “Add” button.

To better assess the relevancy of a shot, the user may click its thumbnail and the corresponding video shot will play in the video player. The user may further invoke a storyboard around that shot (“find more near this one”), or launch a CBIR-board, showing similar shots based on color, texture and layout image features (“find more like this one”). Examples are shown in Figure 4. The user may mark any shot in the results set with positive/negative RF flag. If a shot is found effective as a sample for a CBIR query, the user may also add its number into the query expression.

4.2.2 Search and Browse
Search and browse are complementary in many ways. A search is run by a computer over the entire corpus and provides a ranked list of shots. Browsing is performed by a user who has to eye ball many shots and pick the relevant ones. It usually applies to a local neighborhood of a given sample (e.g., CBIR browsing) or in the close time vicinity of a given shot within a video (a storyboard). While browsing does not require the formulation of a query, it might be ineffective beyond the extent of a local neighborhood. Table 1 summarizes this comparison between search and browse.

Table 1: Complementary properties of search and browse

<table>
<thead>
<tr>
<th></th>
<th>Search</th>
<th>Browse</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Starting point</strong></td>
<td>Query Shot, sample keyframe</td>
<td>Shot, sample keyframe</td>
</tr>
<tr>
<td><strong>Search space</strong></td>
<td>Semantic, rich</td>
<td>Low-level, limited</td>
</tr>
<tr>
<td><strong>Extent</strong></td>
<td>Global</td>
<td>Local neighborhood, “more like/near this”</td>
</tr>
<tr>
<td><strong>Hits order relevant hits</strong></td>
<td>Sorted matches</td>
<td>Low-level sort</td>
</tr>
<tr>
<td><strong>Work is done by</strong></td>
<td>Multiple, ranked</td>
<td>Picked by the user</td>
</tr>
<tr>
<td><strong>Time to perform</strong></td>
<td>Short</td>
<td>Long</td>
</tr>
</tbody>
</table>

Browsing and marking the correct matches may increase recall by elevating the marked shots to the top of the matches list. Taken to extreme, manual browsing through the entire collection would result with finding all the relevant shots at the cost of intensive labor. Efficient search might save a lot of time, however formulating a good query is a task consuming process as well.

4.2.3 The Principles of Mutual Feedback and Session Consistency
During a search session the user iterates the query, searches, browses and provides relevance feedback multiple times. In order to keep the process from diverting away, it is necessary to make the system’s behavior consistent and predictable to the user. For example, the user may mark some shots in a storyboard as positive RF. However, some of the marked RF shots may not be found among the result list for the consequent search of the revised query. This situation is illustrated in the Van diagram shown in Figure 5(Left). Are those shots “lost” in subsequent iterations? If not, how could the user see those “missing” RF shots? These correspond to the little inner RF subsets marked with “+” signs inside the browsable CBIR-board and the storyboard sets of shots.

To keep the new results consistent with all the RF marked by the user in the course of a search session, the system renders the search results in a list composed of three parts Figure 5(Right). The first part is the new search results, ordered by relevancy to the current query and marked by background color according to the session’s accumulated RF. As shown also in Figure 2, shots which were marked as “positive” are rendered with a green background, Shots which were marked as “negative” are rendered with a red background, and all other shots are left with a neutral background.
Search runs are divided to Manual search and Interactive search. In an Interactive run, the user may search the Test set, browse the results and refine his query as desired, mark relevant shots and elevate them. In a Manual run, the user may use a training set to form the query but must avoid interaction with the Test set. After forming the query, it has to be run on the test set without any further human intervention. Hence Manual search does not allow any shots elevation. In both cases the user has up to 15 minutes per topic to formulate and run the query. An Interactive run best exploits the search system functionality, interactivity and browsing capabilities. A Manual run, on the other hand, evaluates better the system’s and query’s ability to generalize to a new Test set, unseen before.

Four system runs were evaluated at TRECVID 04. In all of them, MRF was used during query formulation. The Mean Average Precision (MAP) results are reported in Table 2. The system runs are compared with all other 50 Manual runs and 62 Interactive runs in Figure 6 and Figure 7, respectively. All search runs represent a total of 16 different search systems by participating teams from around the globe. The MRF system performed well in both tasks, demonstrating both rich query expressiveness and generalization at the Manual task and efficient browsing in the Interactive task. In the Manual search our system was outperformed by only one other system, which use a question-answering system, deploys Natural Language Processing and Google search results for query expansion and uses speaker identification with relevance feedback to find specific people [4].

**Figure 5:** Left: Three large circles illustrate three subsets of shots, obtained and visited by search and browse. RF may be marked in any of those subsets. Right: The MRF output starts with the search top matches, followed by shots already marked as “positive” (green background) and “negative” (red background) that were not part of the search top matches. This three parts list is consistent with all previous search & browse iterations.

This colors code provides visual feedback to the user about the quality of the new query results. If the new query is good, much of the green shots will move up the results list and the red ones will move down. If the opposite case is observed, the new query is probably less good than the previous one. Hence the system uses the RF to provide feedback to the user about the quality of the current query given all the RF.

The second and third parts of the list contain all those “lost” RF shots – relevance feedback shots which did not come up in the result list of the new search. They are split into the positive and the negative RF subsets, and are complementary to what is already included above. Hence the user does not lose any manually marked shot, is able to get back feedback about his past selections, and even modify those RF labels if desired.

When the search result is committed as final, a shots elevation process takes place. All the shots marked by the user as “positive” are pushed to the top of the results list, and all the shots marked as “negative” are pushed to the bottom of the list. This shots elevation process conveys a direct contribution of the browsing process to the overall quality of the search result. As seen from the evaluation results, shot elevation has a significant impact on the search results.

**5. EVALUATION**

Evaluation was carried at the official TRECVID 2004 benchmark evaluation, held by the US National Institute of Standards and Technology (NIST). The 24 evaluated topics, defined by NIST and kept hidden from the participants until the development of search systems is frozen, includes a variety of topics; specific people (e.g., Bill Clinton), places (e.g., the Capitol Dome), classes of objects (umbrellas, bicycles, …) as well as generic concepts (cars and pedestrian), actions (people walking up/down stairs) and scenes (water floods). For each topic, the top 1000 retrieved shots were evaluated by Average Precision using a pooled ground truth computed from all the submitted runs [16]. Since SBF ranks video segments and not shots, a simple conversion from segments to shots was applied, giving all the shots within a segment the segment’s score, and imposing internal ranking by shots order.

![Figure 6: Manual search runs at TRECVID 2004. The two Manual MRF Manual runs are marked by solid bars.](image)

In general, Interactive runs generate much better results and higher MAP scores than Manual runs. The advantage of Interactive runs is in potentially better query formulation and in browsing and shot elevation as side effect of the RF process. The user has to decide how much of the 15 minutes to devote to query formulation and how much to shots elevation.

Let $AP(n)$ denote the average precision of a search topic after browsing through the results list to depth $n$, marking all the correct shots as “positive” and elevating those to the top of the results list. By definition, $AP(0)$ is equal to the AP of the search result without elevating any shot, and $AP(n)$ is a non-decreasing function of $n$. 

![Figure 7: Interactive Search runs at TRECVID 2004. The two Interactive MRF runs are marked with solid bars.](image)
Table 2: Mutual Relevance Feedback system runs at TRECVID 2004 Search task, compared to best, average and median of all participating runs.

<table>
<thead>
<tr>
<th>Run</th>
<th>Interactive</th>
<th>Manual Multimodal</th>
<th>Manual Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF System</td>
<td>0.2960</td>
<td>0.1087</td>
<td>0.1056</td>
</tr>
<tr>
<td>Best</td>
<td>0.3519</td>
<td>0.1238</td>
<td>0.1238</td>
</tr>
<tr>
<td>Average</td>
<td>0.1879</td>
<td>0.0644</td>
<td>0.0644</td>
</tr>
<tr>
<td>Median</td>
<td>0.1683</td>
<td>0.0721</td>
<td>0.0721</td>
</tr>
</tbody>
</table>

Table 3: Average Precision (AP) per topic, for the three MRF runs. Topics are sorted in decreasing Manual AP order.

<table>
<thead>
<tr>
<th>Topic #</th>
<th>Topic’s main subject</th>
<th>Manual Multimodal</th>
<th>Manual baseline</th>
<th>Interactive search</th>
</tr>
</thead>
<tbody>
<tr>
<td>134</td>
<td>Bushes Yeltsin</td>
<td>0.414</td>
<td>0.431</td>
<td>0.864</td>
</tr>
<tr>
<td>130</td>
<td>Hockey (goal visible)</td>
<td>0.392</td>
<td>0.412</td>
<td>0.819</td>
</tr>
<tr>
<td>135</td>
<td>Sam Donaldson</td>
<td>0.355</td>
<td>0.198</td>
<td>0.567</td>
</tr>
<tr>
<td>137</td>
<td>Benjamin Netanyahu</td>
<td>0.176</td>
<td>0.117</td>
<td>0.298</td>
</tr>
<tr>
<td>125</td>
<td>Water flood</td>
<td>0.173</td>
<td>0.197</td>
<td>0.352</td>
</tr>
<tr>
<td>138</td>
<td>Golf ball falls into a hole</td>
<td>0.162</td>
<td>0.141</td>
<td>0.134</td>
</tr>
<tr>
<td>128</td>
<td>Money Hyde</td>
<td>0.116</td>
<td>0.124</td>
<td>0.450</td>
</tr>
<tr>
<td>133</td>
<td>Saddam Hussein</td>
<td>0.115</td>
<td>0.170</td>
<td>0.596</td>
</tr>
<tr>
<td>142</td>
<td>Tennis</td>
<td>0.088</td>
<td>0.088</td>
<td>0.189</td>
</tr>
<tr>
<td>145</td>
<td>Horses</td>
<td>0.082</td>
<td>0.077</td>
<td>0.342</td>
</tr>
<tr>
<td>148</td>
<td>Bannons</td>
<td>0.079</td>
<td>0.075</td>
<td>0.218</td>
</tr>
<tr>
<td>147</td>
<td>Building on fire</td>
<td>0.069</td>
<td>0.065</td>
<td>0.149</td>
</tr>
<tr>
<td>127</td>
<td>Person walks dog</td>
<td>0.069</td>
<td>0.119</td>
<td>0.223</td>
</tr>
<tr>
<td>140</td>
<td>Bicycles</td>
<td>0.054</td>
<td>0.007</td>
<td>0.369</td>
</tr>
<tr>
<td>141</td>
<td>People with umbrellas</td>
<td>0.037</td>
<td>0.031</td>
<td>0.148</td>
</tr>
<tr>
<td>144</td>
<td>Bill Clinton</td>
<td>0.033</td>
<td>0.034</td>
<td>0.262</td>
</tr>
<tr>
<td>131</td>
<td>Typing on a keyboard</td>
<td>0.030</td>
<td>0.025</td>
<td>0.157</td>
</tr>
<tr>
<td>143</td>
<td>Wheelchair</td>
<td>0.025</td>
<td>0.063</td>
<td>0.324</td>
</tr>
<tr>
<td>139</td>
<td>Weapon firing</td>
<td>0.016</td>
<td>0.025</td>
<td>0.122</td>
</tr>
<tr>
<td>132</td>
<td>Carrying a stretcher</td>
<td>0.007</td>
<td>0.018</td>
<td>0.143</td>
</tr>
<tr>
<td>125</td>
<td>Pedestrians and cars</td>
<td>0.006</td>
<td>0.005</td>
<td>0.028</td>
</tr>
<tr>
<td>138</td>
<td>Walking up/down stairs</td>
<td>0.004</td>
<td>0.004</td>
<td>0.075</td>
</tr>
<tr>
<td>129</td>
<td>Zoom on capital dome</td>
<td>0.003</td>
<td>0.003</td>
<td>0.079</td>
</tr>
<tr>
<td>Total MAP</td>
<td></td>
<td>0.1087</td>
<td>0.1056</td>
<td>0.2962</td>
</tr>
</tbody>
</table>

We can compute $AP(n)$ for a “baseline” search engine which returns a list of $K$ randomly selected shots from a corpus of $N$ shots. Let $|GT|$ denote the total number of correct shots in the ground truth, then the expected value $\overline{AP(n)}$ is

$$\overline{AP(n)} = \frac{1}{|GT|} \left( np \cdot 1.0 + p \sum_{i=0}^{n} \frac{i}{p} \right)$$

$$\overline{AP(n)} = \frac{1}{|GT|} \left( n(p-p^2) + p^2 K \right)$$

where $p = \frac{|GT|}{N}$

For a given $k$, the expected $\overline{AP(n)}$ grows linearly with $n$, where $0 \leq n \leq k$. A better than random search engine would bring more correct shots to the top of the list, decreasing the relative contribution of browsing and shot elevation. The ideal search engine produces $AP(n)=1.0$ for any $0 \leq n \leq k$, independent of shot elevation which would add nothing. Any practical search engine would result with a non-decreasing $\overline{AP(n)}$ function in between.

The graphs in Figure 8 show $AP(n)$ for each of the 24 search topics (right) and the total $\overline{MAP(n)}$ (left) for $k=1000$. It was computed by taking the search result list of the final query in the Interactive search and the ground truth provided by NIST and computing $AP(n)$ for each value of $n$ (by definition). Notice several “hard” topics for which the graph grows nearly linearly across the entire range of $n$, resembling a random-like results list. For other, “easier” topics, $AP(n)$ shows a rapid increase at low $n$, quickly reaching a flat maximal level when $n$ increases (i.e., no more correct matches are found down the results list). The graph of $\overline{MAP(n)}$ (left) shows a significant increase of the run’s score by browsing and elevating correct shots. At $n=100$ it nearly doubles the MAP score of the search. Hence browsing has a very significant role in the system overall performance.

Figure 8: Simulated effect of shot elevation on total MAP (left) and individual topics (right). A deeper level of marking of “positive” shots in the search results, denoted by the x-axis, increases the result precision and Average Precision (y-axis).

6. CONCLUSIONS AND SUMMARY
In this work, relevance feedback is used for query formulation through an iterative query refinement process. Based on the relevant shots marked by the user, the system identifies useful search terms and allows the user to add them to the query. The effectiveness of this process is supported by experimentation with the Manual search task at TRECVID.

In an Interactive search task, where the user is allowed to interact with the test set, browsing is shown to be equally important to search. Efficient browsing and marking of positive shots serves not only the purpose of RF for query improvement but also as a direct increase of system’s MAP via shots elevation. In our Interactive run, more than half of the overall 15 minutes per topic search was devoted to RF marking and shot elevation.

A lot of attention was given to the design of the user interface and the user interaction. Fine user control on the query refinement process was preferred over a more automatic RF process, providing more flexibility and potentially better search results to experienced users. User’s feedback is complemented by mutual system feedback using color codes and a result list composed of three parts to carry the accumulated RF throughout the iterative search session. Keeping the marked RF shots within the search results provides important feedback to the user regarding the relative quality of the refined query. This mutual feedback is missing in other systems which keep the RF in a dedicated RF area, excluding it from the search results list of subsequent refined searches.

The task of multimodal query formulation in video retrieval remains a great challenge even for an expert user using a flexible and interactive system like the one described here. Several of the TRECVID topics could not be efficiently addressed using the current system, despite of the multiple indices and finely controlled search. Future work should focus first on improving the query representation, the search capabilities and then automating that process.
7. ACKNOWLEDGMENTS

We would like to thank NIST for organizing the TRECVID benchmark, LDC, CMU, CLIPS-IMAG, and the Collaborative Annotation forum for providing much of the training and metadata needed for the search task, and especially our colleagues in the IBM team at TRECVID for providing us with the semantic concepts and CBIR indices for the 2004 Search task evaluation.

8. REFERENCES


http://www-nlpir.nist.gov/projects/trecvid/


