Interactive Multimedia Search: Systems for Exploration and Collaboration

Jeremy Pickens
John Adcock
Matthew Cooper
Maribeth Back
Pernilla Qvarfordt
Gene Golovchinsky
Andreas Girgensohn

This work is based on earlier works:

Algorithmic mediation for collaborative exploratory search, in Proceedings of SIGIR '08 ©ACM, 2008, Pages 315–322. http://doi.acm.org/10.1145/1390334.1390389

Experiments in interactive video search by addition and subtraction, in Proceedings of CIVR '08 ©ACM, 2008, Pages 465–474. http://doi.acm.org/10.1145/1386352.1386412

Abstract

We have developed an interactive video search system that allows the searcher to rapidly assess query results and easily pivot off those results to form new queries. The system is intended to maximize the use of the discriminative power of the human searcher. The typical video search scenario we consider has a single searcher with the ability to search with text and content-based queries. In this paper, we evaluate a new collaborative modification of our search system. Using our system, two or more users with a common information need search together, simultaneously. The collaborative system provides tools, user interfaces and, most importantly, algorithmically-mediated retrieval to focus, enhance and augment the team's search and communication activities. In our evaluations, algorithmic mediation improved the collaborative performance of both retrieval (allowing a team of searchers to find relevant information more efficiently and effectively), and exploration (allowing the searchers to find relevant information that cannot be found while working individually). We present analysis and conclusions from comparative evaluations of the search system.

1 Introduction

The infrastructure and technology for maintaining large digital video collections has reached a point where use and distribution of these assets is commonplace. However, the ability to search within the audio-visual content of such collections remains relatively primitive The success of video management systems rest on the integration of two evolving technologies: video content analysis, and interactive multimedia retrieval. Video analysis systems derive content-based indices of the video data, and interactive information retrieval systems allow searchers to navigate those indices to identify content that satisfies some putative information need.

Over the last several years our group has developed an interactive video search system, named MediaMagic [2, 13, 12], designed to enable users to efficiently assess search results using a flexible interface and rich visualizations. Searches can be performed using textual, visual, and semantic content. Results are displayed in rich query-dependent summary visualizations that encapsulate the relationship between the query and the search result in multiple visual dimensions. As the user steps into the search results these cues are maintained in the various representations of keyframes and timelines, along with visual cues to indicate navigation history and existing relevance judgments. Search by example is enabled throughout to flexibly explore the search space.

The video information retrieval problem is the focus of a growing research community which includes the TRECVID evaluations [26, 30]. We have participated in TRECVID since 2004 and its protocol forms the experimental framework used in the experiments described below. This community has produced a rich variety of semantic content analysis techniques and associated interactive search systems that have steadily advanced interactive search performance [10, 11, 18, 21, 22, 28, 33, 32].

MediaMagic has achieved state of the art performance in video retrieval evaluations by combining proven content analysis techniques with a powerful user interface. Collaboration between multiple searchers is a complementary means to further advance search systems. We focus this paper on extending our search system to enable collaboration. It is intuitive that information seeking can be performed more effectively as a collaboration than as a solitary activity. Different people bring different perspectives, experiences, expertise, and vocabulary to the search process. A retrieval system that takes advantage of this breadth of experience should improve the quality of results obtained by its users [6].

We distinguish several kinds of collaboration in the context of information retrieval. Collaborative filtering is an example of asynchronous and implicit collaboration; aggregate crowd behavior is used to find information that previous users have already discovered [31]. The term "collaboration" has also been used to refer to synchronous, intentionally-collaborative information seeking behavior. Such systems range from multiple searchers working independently with shared user interface awareness [24] to multiple people sharing a single user interface and cooperatively formulating queries and evaluating results [29]. Collaborative web browsing and real-time sharing of found information, through specialized user interfaces rather than through email, is common in these systems [36]. A major limitation of existing synchronous approaches is that collaboration is restricted to the user interface. Searchers are automatically notified about the ongoing activities of their teammates, but to exploit that information to improve their searches, each user must manually examine and interpret teammates' queries and found documents. While awareness of one's co-searcher(s) is an important first step for collaborative retrieval, user interface-only solutions still require too much attention to others' results.

The synchronous, intentional approaches mentioned above are closely related to the system described in this paper. Figure 1 shows the structural differences in architecture between three user interface-only collaborative IR systems ([6], [29], and [24]) and the algorithmically mediated approach. All three earlier systems use search engines that are not aware of the ongoing collaboration. As each query arrives at the engine, it is treated as a new, separate search. Although searchers may collaborate at the user interface and interpersonal level, the search engine itself does not support collaboration. This is true whether each searcher uses a separate search engine, or if they share a search engine as in Físchlár-DiamondTouch [29]. In SearchTogether [24], a searcher's activity is not used by the underlying engine to influence the partner's actions; all influence happens in the interface or live communication channels. In contrast, an algorithmically-mediated collaborative search engine coordinates user activities throughout the session.

We describe a retrieval system wherein searchers, rather than collaborating implicitly with anonymous crowds, collaborate explicitly (intentionally) with each other in small, focused search teams. Collaboration goes beyond the user interface; information that one team member finds is not just presented to other members, but it is used by the underlying system in real-time to improve the effectiveness of all team members while allowing each to work at their own pace.

In this paper, we present comparative results from the latest evaluations of single-user and collaborative versions of our interactive video search system. The multi-user variant of our search system integrates algorithmic mediation and intentional collaboration. The design comprises a set of user interfaces, a middleware layer for coordinating traffic, and an algorithmic backend optimized for collaborative exploratory search. We evaluated the effect that algorithmic mediation has on collaboration and exploration effectiveness. Using mediated collaboration tools, searchers found

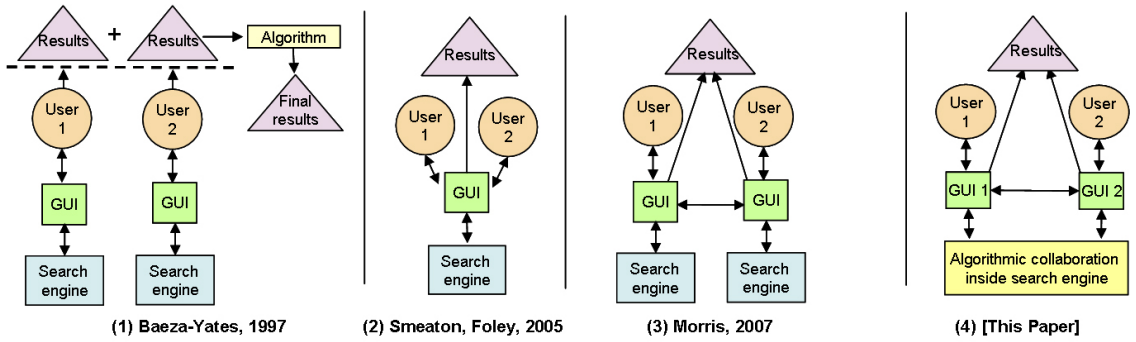

Figure 1: System architecture comparisons.

relevant video more efficiently and effectively than when working individually, and they found relevant video that otherwise went undiscovered.

2 MediaMagic Search System

Our single-user video search system, MediaMagic, comprises analysis, interactive operation, and post-interactive components. Our analysis begins with data pre-processing to generate indices into the video collection. The second component is the search interface by which the searcher navigates the video collection using the various indices. The final component is the post-processing of the user's input to augment the user-identified search results through automatic querying. Much of the MediaMagic system has been documented in detail elsewhere[2, 13, 12, 3]; we include an overview here for completeness.

2.1 Data pre-processing

2.1.1 Segmentation

The bootstrapping data-processing step is video shot boundary determination. Given a shot-level segmentation, we identify higher-level topic or story units to augment the shot boundaries. We compute the new boundaries with a novelty-based segmentation of the text transcripts in a latent space as described in [3]. These longer story segments are the primary unit of retrieval during search.

2.1.2 Text Indexing

In preparation for interactive operation text indices are built for both the shot-level and story-level segmentations using Lucene [23] (for keyword search) and our latent semantic indexing system (for fuzzy text search). For the latent space, we build a latent space [8] of fixed dimension 100 using the sparse vectors of stopped and stemmed terms.

2.1.3 Visual and Semantic Indexing

Color correlograms [19] are computed for each shot's keyframe and used for computing visual similarity during interactive operations. In addition, visual features are extracted for use in the semantic indexing described below. We extract YUV color histograms for each frame as follows. We compute 32-bin global frame histograms, and 8-bin block histograms using a 4×4 uniform spatial grid for each channel. We select keyframes from each shot in the reference segmentation by minimizing the chi-square distance between each frame histogram and the centroid for the shot. Finally, SURF descriptors with their vertical pixel (Y-axis) location are also computed [7] and quantized into 200 bins using online k-means [9]. The quantized SURF descriptors are used together with the keyframe's color histogram features for semantic concept detection.

MediaMagic includes the ability to search with using semantic similarity measurements. We construct single concept detectors for the LSCOM-Lite concept set [25] using support vector machines (SVMs). We use reduced training sets for parameter tuning and train our detectors using asymmetric bagging [35] following the procedure detailed in [4]. After training the SVMs, we combine their probabilistic output predictions by averaging. This approach achieved mean average precision of 0.251 for the LSCOM-Lite concept set on the MediaMill benchmark training and test sets [34]. For indexing, each shot has an associated 35 element vector describing the posterior probability of each of the high-level concepts. For the semantic distance between two shots we use the mean absolute distance (normalized L1) between their respective concept vectors.

2.2 Search Engine and Interface

The video search system interface is pictured in Figure 2. The search topic description and supporting examples images are shown in area C. Text and image search elements are entered by the searcher in area B. Search results are presented as a list of story visualizations in area A. A selected story is shown in the context of the timeline of the video from which it comes in area E and expanded into shot thumbnails in area F. When a story or shot icon is moused-over an enlarged image is shown in section D. When a story or shot video segment is played it is also shown in

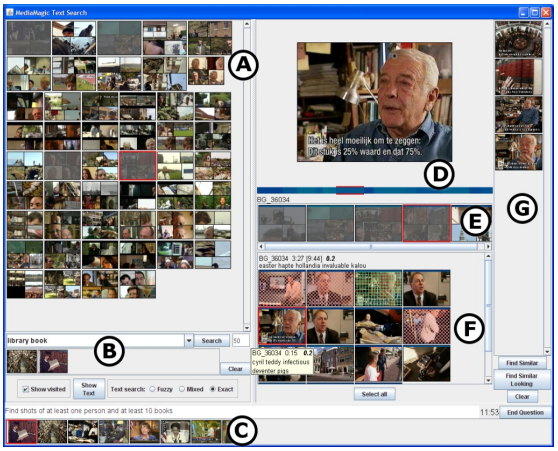

Figure 2: Interactive system interface. (A) Search results area with story keyframe summaries. (B) Search text and image entry. (C) Search topic and example media. (D) Media player and keyframe zoom. (E) Story timeline. (F) Shot keyframes. (G) Relevant shot list.

area D. User selected shot thumbnails are displayed in section G.

2.2.1 Text Query

The searcher can choose an exact keyword text search, a latent semantic analysis (LSA) based text search, or a combination of the two where the keyword and LSA-based retrieval scores are averaged. We use the text transcript (when available; see section 2.3) to provide text for story and shot segments. The exact text search is based on Lucene [23] which ranks each story based on the tf-idf values of the specified keywords. In this mode the story relevance, used for results sorting and thumbnail scaling and color coding as described in following sections, is determined by the Lucene retrieval score. When the LSA based search is used [8], the query terms are projected into a latent semantic space (LSS) of dimension 100 and scored in the reduced dimension space against the text for each story and each shot using cosine similarity. In this mode, the cosine similarity determines the query relevance score. In our application the LSS was built treating the text from each story segment as a single document. When determining text-query relevance for shots, each shot gets the average of the retrieval score based on the actual shot text and the retrieval score for its parent story. That is, the shots garner some text relevance from their enclosing story.

2.2.2 Image Query

Any keyframe in the interface can be dragged into the query bar (Figure 2 B)and used as part of the query. Each query shot's color correlogram is compared to the correlogram for every shot thumbnail in the corpus. The maximum image-similarity score from the component shots is propagated to the story level. The document scores from the text search and image similarity are combined to form a final overall score by which the query results are sorted.

The searcher may also take any selection of shots and stories in the interface and perform a "find similar looking" operation (accessed through a context menu). In this operation the selected shots are used to perform an image-based search using color correlograms. It is equivalent to putting all the selected shots in the image-query area and clearing the text search box, but being much simpler to perform provides a significant shortcut.

2.2.3 Concept query

A searcher can alternatively choose to perform a "find similar" operation on a set of selected shots and stories. Two similarity measures are combined to order stories for retrieval. The similarity between the text of the selected segment(s) and those of candidate stories is combined with the similarity between the concept vectors of the selected segments and those of candidate stories. The text-similarity is the cosine distance between the text (in latent space) of the selected segment(s) and the text of each candidate segment. The concept distance is the minimum distance between the concept vectors of the example shots and the concept vectors of each candidate segment. The two similarity scores are averaged together to create a similarity score for each candidate segment.

2.2.4 Visual cues

The search engine performs retrieval of story segments based on the indices described above. Shots are represented with a single thumbnail. Stories are represented with a query-dependent summary thumbnail; the thumbnails from the four highest scoring shots against the current query are combined in a grid. The area allotted to each shot in this four image montage is proportional to its relative retrieval score.

Semi-transparent overlays are used to provide three cues. A gray overlay on a story icon indicates that it has been previously visited (see Figure 2 A and E). A red overlay on a shot icon indicates that it has been explicitly marked as not-relevant to the topic (see Figure 2 F). A green overlay on a shot icon indicates that it has been explicitly marked as relevant to the topic (see Figure 2 F). A horizontal colored bar is used along the top of stories and shots to indicate the degree of query-relevance, varying from black to bright green. The same color scheme is used in the timeline depicted in Figure 2 D.

An optionally displayed dialog provides information about the underlying transcript and text query operation. The dialog shows the transcript from the selected shot or story along with terms related to the query (determined from the latent semantic space) and indicates query terms that are not contained in the dictionary. Also the entire dictionary is displayed

in a scrolling window allowing the user to browse the available terms.

2.2.5 Post-Interactive Processing

When the interactive search session on a particular topic ends, the search system automatically extends the user-selected shots with an automatic process. First, the shots neighboring (or bracketing) the user-identified relevant shots are added to the result list (even if they were marked as not relevant by the user). Next, the text from the shots that have been judged by the searcher to be relevant is combined to form a single LSA-based text query. This query is applied to the unjudged shots and the highest scoring ones retained for the result list. Finally, the concept vector of every unjudged shot is compared against the concept vectors of the judged shots. For each group (relevant, not-relevant) the minimum distance is computed, yielding a positive and negative similarity measure for each unjudged candidate shot. After bracketing, the remaining unjudged shots are ranked by an equal weighting of semantic similarity and text similarity to form an ordering from which to select likely shots.

2.3 Text-free Search

We have implemented a version of our system that makes no use of text during pre-processing or search. This was accomplished by altering the system described in the previous section in several small ways. To determine a story-level segmentation we use the semantic concept vectors (from section 2.1.3) for each shot instead of the MT/ASR transcript. We use the same novelty-based segmentation of [3] but instead of measuring the inter-shot novelty of the transcript we measure the novelty of the concept vectors using the same cosine distance metric. Story boundaries are placed at points of high inter-shot concept novelty. In this way we preserve the story-and-shot multi-level indexing structure used in the basic interactive system without falling back to a fixed segmentation. Next, we disabled the text query box and the text-based similarity searching of section 2.2.3 by building a text index with empty transcripts. Query relevance in the text-free system is determined solely by similarity between the color-correlograms (with the "find similar looking" and image query operations), and semantic concept vectors (with the "find similar" operation).

3 Multiple User Collaboration

We have incorporated the MediaMagic core into a collaborative search system which comprises a set of interfaces and displays, a middleware layer for handling traffic, and an algorithmic engine optimized for collaborative exploratory search. By interacting with each other through system-mediated information displays, searchers help each other find relevant information more efficiently and effectively. Each searcher on a team may fill a unique role, with appropriately optimized interface and display components. Query origination, results evaluation, and results partitioning are examples of such roles. The design of the collaborative system was based partially on lessons learned from best-of-breed instances of video search interfaces [17], and partially from observations and studies we performed as part of the design process. The TRECVID search task provides an interesting and complex search task in which to evaluate our collaborative ideas while leveraging existing expertise.

We emphasize that the underlying mediation algorithms supporting this task are generic and may be applied to all types of retrieval: text, video, images, music, etc. Our search engine algorithmically mediates a wide variety of queries, including text queries, fuzzy text queries (text-based latent semantic concept expansion), image similarity queries based on color histograms, and image-based concept similarity, via statistical inference on semantic concepts of images.

In this section we will describe how the system combines multiple iterations from multiple users during a single search session. The system consists of three parts: (1) user interfaces that implement the roles, (2) the architecture to support these roles, and (3) algorithms used to perform collaborative search.

3.1 Search Roles

The synchronous and intentional nature of the collaboration enables searcher specialization according to roles and/or tasks. Many roles and associated task types are possible; these may shift over time or during different parts of the search task. Roles may be equal, hierarchical, partitioned (separated by function), or some combination thereof. User interfaces, tools, and algorithms may offer commands or perform actions specific to particular roles.

Our current system allows collaborating users to assume the complementary roles we dubbed Prospector and Miner. The Prospector opens new fields for exploration into a data collection, while the Miner ensures that rich veins of information are explored. These roles are supported by two different user interfaces and by underlying algorithms that connect the interfaces. Unlike approaches in which roles are supported manually or only in the user interface [24], these roles are built into the structure of the retrieval system. The regulator layer (described below) manages roles by invoking appropriate methods in the algorithmic layer, and routing the results to the appropriate client.

3.2 Collaborative System Architecture

The collaborative system architecture consists of three parts: the User Interface Layer, the Regulator Layer, and the Algorithmic Layer (Figure 3). System compo-

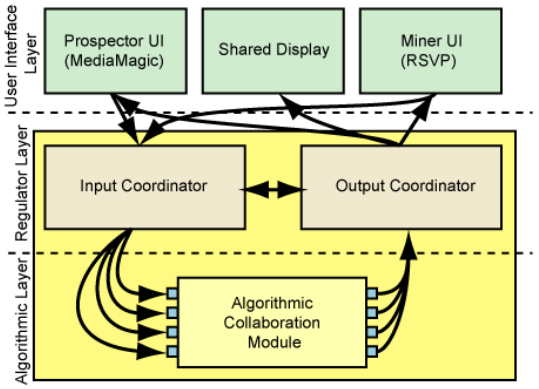

Figure 3: The collaborative system architecture.

nents communicate through a web service API, and can be combined in different ways: the single shared display in a co-located setting can be replaced by separate displays in remote locations, showing the same information.

3.2.1 User Interface Layer

Our system contains three user interfaces: (1) A rich query user interface (MediaMagic [1, 14]) for use of the Prospector, (2) a rapid serial visualization result browsing user interface (RSVP) for use of the Miner, and (3) a shared display containing information relevant to the progress of the search session as a whole.

The MediaMagic user interface (as described in section 2) contains flexible tools for issuing queries (text, latent semantic text, image histogram, and concept queries), displays ranked results lists and has an area for viewing and judging retrieved shots. The RSVP user interface is primarily designed for relevance assessment of video shots, which are presented in a rapid but controllable sequence. However, the RSVP user interface also includes the capability for Miners to interrupt the flow of shots to issue their own text queries.

Finally, a shared display shows continually-updating information about issued queries, all shots marked as relevant by either user, and system-suggested query terms based on activities of both users. In our setting, the shared display was shown on a large screen easily viewed by both the Prospector and the Miner (Figure 5, top center).

3.2.2 Regulator Layer

The regulator layer consists of an input regulator and an output regulator. The input regulator is responsible for capturing and storing searcher activities, such as queries and relevance judgments. It contains coordination rules that call the appropriate algorithmic collaboration functions. The input regulator implements policies that define the collaborative roles. Similarly, the output regulator accepts information from the algorithmic layer and routes it to appropriate clients based on their roles. The regulator works autonomously, and does not interact directly with users.

Figure 4: Shared display interface (top) and RSVP interface (bottom).

3.2.3 Algorithmic Layer

The algorithmic layer consists of a number of functions for combining searchers' activities to produce documents, rankings, query suggestions, and other information relevant to the search. It performs basic searches, and generates raw search results, transformed search results based on input from multiple users, and query terms that characterize the current state of the collaboration. Details of these algorithms are discussed in the following section.

3.3 Algorithmic Mediation

We define two weighting variables, relevance $w_r(L_k)$ and freshness $w_f(L_k)$. These are functions of L_k, a ranked list of documents retrieved by query k.

$$\begin{aligned} w_f(L_k) &= \frac{|unseen \in L_k|}{|seen \in L_k|} \\ w_r(L_k) &= \frac{|rel \in L_k|}{|nonrel \in L_k|} \end{aligned} \quad (1)$$

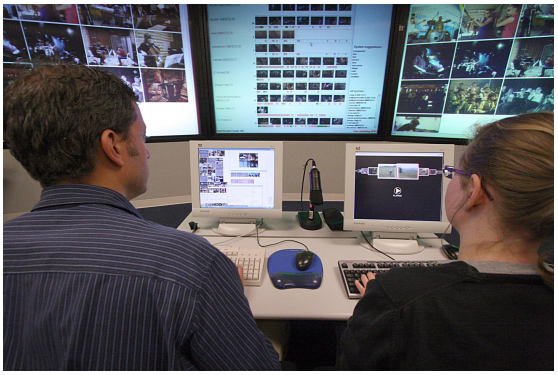

Figure 5: A collaborative search session. Each user's UI is suited to their role: Prospector (left) and Miner (right). Large side screens show sample relevant shots for the current topic; center screen shows the shared query state.

The query freshness weight w_f is given by the ratio of unseen (retrieved by the engine, but not yet manually examined) to seen (retrieved and manually examined) documents in L_k. The query relevance weight w_r is given by the fraction of seen documents that were judged relevant for that query. These two factors are designed to counterbalance each other. If a query retrieved many relevant documents, it should have a high relevance weight, but once most of the documents from a query have been examined, other queries should have higher priority given by the freshness weight. These weights are updated continuously based on searchers' queries and judgments of relevance. The weights are then used to affect the information shown to each searcher, as appropriate to their roles.

3.3.1 Miner Algorithm

As mentioned in section 3.1, the RSVP client acts in the role of Miner. The regulator accumulates documents retrieved by all team members during a session. Documents not yet examined by the Prospector are queued for the Miner based on freshness and relevance weights. The queue is ordered by a score that uses Borda fusion to merge the contributions of all queries, as shown in Equation 2.

$$score(d) = \sum_{L_k \in \{L\}} w_r(L_k) w_f(L_k) borda(d, L_k) \quad (2)$$

The Prospector continually adds new ranked lists L_k to the set L, and views and judges documents. Meanwhile, the Miner judges highly-ranked unseen documents d. These documents are likely to have appeared in more than one list L_k; therefore, relevance judgments made on these documents affect the w_f and w_r weights of more than one list and further change overall priorities. The Miner does not have to manually decide which documents to comb through, nor does the Prospector have to decide which documents to feed to the Miner.

The Miner algorithm is similar to the on-line hedge algorithm for ranked list fusion [5]. Both approaches share the intuition that attention should shift to those lists that show themselves to be more "trustworthy." In our work, however, the ranked lists being combined are from different queries, rather than from different search engines. Furthermore, the number of ranked lists is not static. As both users issue queries, the number of rank lists grows over time.

3.3.2 Prospector Algorithm

The previous section describes how algorithmic mediation allows the Miner to work with the Prospector. But how does the Prospector algorithmically influence the Miner? The unseen documents priority score in Equation 2 only affects the ordering of documents for the Miner. We chose not to apply the same transformation to the Prospector's search results because we wanted the Prospector to see the raw effectiveness of each query. If previously retrieved but unseen documents were retrieved again by a new query, that would boost their priorities in the Miner's queue; those documents would likely soon receive attention by the Miner rather than the Prospector.

We choose to let the Prospector focus on coming up with new avenues for exploration into the collection. This is accomplished by a real-time query term suggestion feed from which the Prospector can get a sense of how the overall search is progressing and draw ideas about new avenues to explore. The basic idea is similar to the Miner algorithm. However, instead of a Borda count on unseen documents d, we use a "ranked list frequency" count on terms t: $rlf(t, L_k)$. This is defined as the number of documents in L_k in which t is found. Using rlf, we define a score for every term, t:

$$score(t) = \sum_{L_k \in \{L\}} w_r(L_k) w_f(L_k) rlf(t, L_k) \quad (3)$$

This function updates continuously. Terms used in previous queries are filtered out to produce a list of top ten terms that are shown on the shared display. As the Miner's activity affects the w_r and w_f weights, the system re-orders or replaces term suggestions. The more the Miner digs into fresher and more relevant pathways, the more terms associated with those pathways appear. Once a particular avenue loses freshness or does not exhibit relevance and the Miner switches from that path, the automatically-suggested terms switch accordingly. No explicit actions are required for the Miner to suggest terms to the Prospector, just as no explicit actions are required from the Prospector to feed documents to the Miner; the collaborative retrieval algorithm handles the flow of information between the users. The Miner and Prospector are self-paced in their respective workflows, but the influence that each exerts on the other is synchronous.

4 Experiments

In this section we review experiments conducted in the context of the TRECVID 2007 evaluation [26]. The evaluation comprises 24 multimedia search topics executed over a corpus of roughly 50 hours of Dutch "infotainment" television. Dutch transcripts from automatic speech recognition [20] and machine-translations to English (referred to below as MT/ASR), were provided, along with a common shot-level segmentation [27]. For each topic (specified by a text description and supporting image and video relevant examples) the searcher is given 15 minutes to search for shots which satisfy the topic statement. In this section, we breakdown the performance of our various collaborative, single user, and visual-only systems.

We hypothesized that collaborative search would produce better results than *post hoc* merging (see the Baeza-Yates [6] architecture in Figure 1) with respect to average precision and average recall. We discovered that collaborating users found more unique relevant documents. We also compare the end effects of two-person collaboration against two-person *post hoc* results merging.

4.1 Summary Performance

Mean average precision (MAP) is the principal summary performance metric used in the TRECVID evaluations. Figure 6 shows MAP scores for several system variations. For each system we evaluated the 24 TRECVID 2007 search topics, dividing the labor among 4 searchers. The system denoted SUA (single-user all) is our standard single-user MediaMagic system which includes text search, text similarity, image similarity, and concept similarity search. SUA_b denotes a second trial using the SUA system with a different set of 4 searchers. The variant denoted SUV uses no text information, leaving the searcher to work only with image similarity and concept similarity searches. Visual-only systems are a standard baseline for the manual and automatic search tasks at TRECVID, but are a less common variation among the interactive search systems, due presumably to the human cost of performing interactive evaluations. The CO15 systems employs the real-time, multi-user, collaborative search system described in section 3 and includes text search.

Figure 6 presents a breakdown of the achieved MAP of the 3 single-user and 15 minute collaborative runs. Each score is shown in 3 parts: the MAP achieved by including only shots explicitly identified by the user during the interactive portion, then (as described in section 2.2.5) the MAP achieved by adding the neighboring shots to that result, and finally the MAP achieved by evaluating the complete list of 1000 shots including the automatic search results as described in section 2.2.5. The performance is dominated by the shots identified directly by the user.

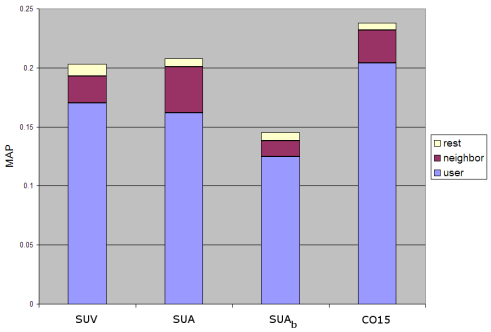

Figure 6: MAP on the 24 TRECVID 2007 topics broken down by contribution from user-selected shots (user), then adding neighboring shots (neighbor), and then the rest of the submitted shots (rest). Note that the visual-only run (SUV) outperforms the visual+text run (SUA) when only user-identified shots are considered, but this difference is not statistically significant.

We evaluated the statistical significance of the measured MAP differences using a standard permutation test [16]. The test measures the chance that a random assignment of average precision scores between two systems yields the same or greater difference in observed MAP.

In our results, the difference with the strongest measured statistical significance is that SUA_b performs reliably worse (significant at p=0.02 or better) than every other system. SUA_b was performed by a less experienced group of searchers (more on this in section 4.2). This user-dependence of the system performance is undesirable, but not unexpected. Meanwhile the 15 minute collaborative system (CO15) outperforms the single-user systems at a statistical significance level of p=0.1 (though not at p=0.05). The measured significance of this comparison rises however when examining only the shots explicitly selected by the user. In this case the observed differences in AP between CO15, and SUA or SUV are significant at p=0.05. There is no statistically significant difference between the with-text and without-text single-user systems, SUA and SUV.

From this analysis it is clear that removing the capacity for text search and substituting our semantic similarity for the text-based similarity during story segmentation does not significantly degrade performance. The with-text SUA and without-text SUV systems are indistinguishable under this summary statistic. The implication is that the correlation between the transcripts and the content of this corpus is fairly weak (or at least no stronger than the correlation with visual and concept features), at least for the tested topics. This can be ascribed to an unknown combination of factors: the nature of this specific video corpus, the nature of the search topics, the quality of the translated transcript, and the quality of the visual and semantic indexing.

4.2 Single Searcher Performance

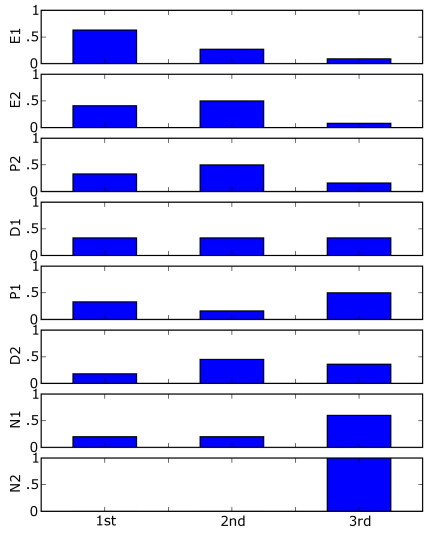

Figure 7: Normalized histograms of searcher average precision ranks across all topics for the 3 standalone runs (SUA, SUV, and SUA_b) sorted by Borda score. In decreasing order of familiarity with the system are: (E)xperts, (D)evelopers, (P)ros, and (N)ovices.

For each of the 24 search topics, we had 3 independent single-user search results. Figure 7 shows for each of the 8 searchers a normalized histogram of that searcher's AP rank accumulated across all performed topics. The searchers are sorted, top to bottom, in order of decreasing Borda score of their rank distributions. The height of the bar for the row labeled N2 and the column labeled 1st is the fraction of the topics answered by user N2 which were ranked 1^{st} among the 3 single user trials on that topic. The searchers are labeled here by experience level: E1 and E2 are "experts" with multiple years of experience developing and using the MediaMagic interface . D1 and D2 are "developers" and searchers of this year's system. Together the 4 "expert" and "developer" users performed the SUA, SUV, and CO15 evaluations. The latter in pairs of 1 "expert" and 1 "developer". P1 and P2 are "pros" who have developed and used the MediaMagic system in previous years but not recently, and N1 and N2 are "novices" who have never used the system before these trials. A spectrum of user-performance is evident with several searchers more likely to place 1^{st}, several searchers more likely to place 3^{rd}, and the rest falling somewhere in between. Note that the "expert" and "developer" users performed more trials in this round of tests (12 topics each on single-user systems as well as another 12 on the collaborative system) than the "pro" and "novice" users who performed only 6 topics each to create the lower-performing SUA_b result. So in addition to having more long-term experience, our more experienced users had the additional benefit of any learning effects that might accrue over the course of testing with the different system variations.

4.3 Multiple User Experiments

For the collaborative experiments, we used a mixed-design method, where teams of searchers performed one of two search conditions (merged or collaborative); all 24 TRECVID interactive retrieval topics were used in both conditions. The teams in the collaborative condition (CO15) consisted of two co-located participants with different levels of experience in multimedia information retrieval; one had prior experience (an aforementioned "expert") and one did not (an aforementioned "developer"). The expert used the MediaMagic client, while the developer used the RSVP client. Prior to the runs, both team members received training on their respective user interfaces, general instruction on their expected roles, and how the system implemented each role. Verbal and non-verbal communication between participants was not controlled; participants were free to discuss the topic as necessary. Each expert user worked and equal amount with each developer user; switching team members balanced the influence of individual differences on team performance. Although it would be interesting to explore role interaction effects, we did not have enough subjects to pursue that analysis.

To provide a comparison with the multi-user collaborative system we formed a merged condition consisting of *post hoc* unions of the SUA and SUA_b search runs with no interaction or algorithmic mediation. This condition mirrored the collaborative condition in terms of the composition of the teams: each topic was completed by a team with one experienced user and one inexperienced user, using the same amount of overall time. All team members received training on, and used, the MediaMagic client. Teams were also swapped to balance individual influences.

In both conditions the same underlying indices and similarity functions were used. The primary difference was that the interfaces in the *post hoc* merged condition were connected to the stand-alone retrieval engine, whereas in the collaborative condition the interfaces were connected to the algorithmically-mediated collaborative engine.

The CO15 and SUA conditions were originally designed for the TRECVID 2007 competition; their results were submitted to the NIST relevance judgment pool. However, the SUA_b run, and therefore the post hoc merge of SUA and SUA_b, was completed later. In these later runs, individual searchers selected a small number of shots that had not been found by any other search system and therefore had not been judged by NIST. In order to avoid incorrectly penalizing these latter runs through incomplete ground truth, four judges independently assessed shots unique to the merged condition for topical relevance. Disagreements among judges were resolved in a joint judging session. In all, 41 relevant shots were added to the ground truth, raising the total number of relevant shots across the 24 test topics from 4704 to 4745.

4.3.1 Metrics

In evaluating the effectiveness of interactive search, we need to distinguish between documents returned by the search engine, documents actually seen by the user, and documents selected by the user [15]. Thus, we use *viewed precision* (P_v, the fraction of documents seen by the user that were relevant), *selected precision* (P_s, the fraction of documents judged relevant by the user that were marked relevant in the ground truth), and s*elected recall* (R_s) as our dependent measures.

The MediaMagic search system provides an automated query step described in section 2.2.5, but in our analysis of interactive performance we only consider relevant shots that were marked explicitly by the user.

We wanted to test the hypothesis that mediated collaboration offers more effective searching than *post hoc* merging of independently produced results, as was done, for example by Baeza-Yates *et al.* [6].

4.3.2 Collaboration Results

To assess the teams' performance, we removed duplicate shots from the merged result set, and kept track of which shots each person saw (whether they judged them or not). Participants in the merged condition saw an average of 2978 distinct shots per topic (both relevant and non-relevant); participants in the collaborative condition saw an average of 2614 distinct shots per topic. For each topic, we subtracted the merged score from the collaborative score and divided by the merged score. We also split runs up over time (3.75, 7.5, 11.25 and 15 minutes).

We found that collaborative search consistently outperformed merged search on our metrics as shown in Table 1 and in Figure 8. For example, at the end of the 15 minute session, $\bar{R}_s$ was 29.7% higher for collaborative search than for merged results. Collaborative search exhibited better performance throughout the session.

	3.75m	**7.5m**	**11.25m**	**15m**
P_s				
Overall	+9.8	+21.5	+22.4	+30.2
Plentiful	-2.6	+6.1	+4.2	+0.4
Sparse	+22.4	+36.8	+40.7	+60.1
$\bar{R}_s$				
Overall	+15.2	+35.7	+19.2	+29.7
Plentiful	+13.9	+13.5	+3.8	-4.4
Sparse	+16.4	+57.9	+34.7	+63.8
P_v				
Overall	+13.6	+65.4	+41.1	+51.1
Plentiful	+16.6	+9.1	+2.3	-9.7
Sparse	+10.6	+121.6	+79.9	+111.9

Table 1: Average percent improvement collaborative over merged, at 3.75, 7.5, 11.25 and 15 minutes.

Overall results of the experiment indicate a consistent advantage for collaborative search (CO15) over merged results from independent single-user searches (SUA + SUA_b). We wanted to understand the differences in more detail, and thus looked at the effect that the number of relevant shots for a topic had on our results. We divided the 24 topics into two groups based on the total number of relevant shots available for that topic. Topics that fell below the median (130) were deemed "sparse" (average of 60 relevant shots per topic) and those above were "plentiful" (average of 332 relevant shots per topic).

For "sparse" topics, users in the merged condition saw on average 3787 unique shots vs. 2877 in the collaborative condition; for "plentiful" topics, users in the merged condition saw on average 2168 shots vs. 2352 for the collaborative condition. We then repeated our analysis on the two groups independently; results are shown in Table 1, and compared in Figure 8.

We now see that for plentiful topics, collaborative search is comparable to merging individual results: the white bars in Figure 8 are small, and the error bars span 0. That is: if relevant content is abundant, anybody can find it. When the topics are not so obvious, however, collaborative search produces dramatically better results on average. The gray bars are consistently above 0 and are consistently larger in magnitude than the solid "overall" bars. For viewed precision (P_v) in particular, we saw an increase of over 100% compared to the merged condition, despite the fact that participants in the merged condition saw 910 more shots overall than in the collaborative condition. A repeated-measures ANOVA of P_v confirmed that this interaction between time and topic sparsity ($F(3, 66) = 3.69, p < 0.025$) was significant, indicating that improvements over the course of a session were unlikely to be due to chance.

Although we did not design the experiment to quantitatively measure the effect of oral communication between team members, we analyzed the video record of experiments to assess the degree to which non-mediated communication channels (e.g. gaze) were used by our participants. We found that the Miner spent on average 5.3 seconds (SD=6.4) of a 15 minute session looking at the Prospector's screen, while the Prospector spent 4.2 seconds (SD=3.7) looking at the Miner's screen. Given that the system's algorithmic mediation, not the Prospector, determined the shots and their order of presentation to the Miner, and given that Miner found on average 38% of the relevant shots, it is unlikely that non-algorithmic channels played a large part in the overall team performance.

These results suggest that algorithmically-mediated collaborative teams were much more efficient than people working individually at detecting relevant content.

4.3.3 Exploration

In addition to effectiveness, we wanted to see how well exploration was supported by algorithmic mediation. We wanted to assess the comparative effectiveness of collaborative versus *post hoc* merged search in finding unique relevant content; content that only one system was able to find. While some retrieval tasks are con-

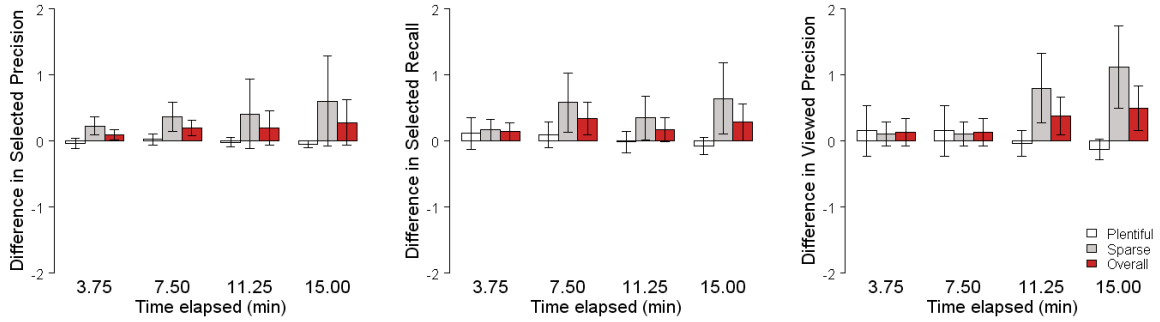

Figure 8: Plentiful/Sparse Split: Selected Precision, Selected Recall, Viewed Precision. Error bars represent ±1 standard error.

sidered a success if any relevant document is found, sometimes it is appropriate to examine *which* relevant documents are found.

Head-to-head comparisons between two systems for uniqueness are problematic. Without an external baseline, uniqueness is equivalent to the difference in the number of relevant documents retrieved. Thus we chose to use the data submitted by the 10 other TRECVID participants as the baseline.

Each TRECVID group was allowed to submit several runs, all of which were used by NIST to determine relevant documents. Some of these runs were produced by parametric variation on the same low-level indexing and retrieval algorithms and retrieved many of the same documents. Therefore, to get a more accurate count of unique documents, we kept only the best-performing (in terms of MAP) run from each group. This gave us a "background" set of 10 runs, the union of which served as our baseline.

Our two runs, collaborative and *post hoc* merged, share the same low-level indices and document similarity functions. Therefore, we compare each condition separately against the baseline. We first evaluate the collaborative system against the baseline by computing for each topic N_u the number of unique relevant documents identified by each of the 11 systems: 1 collaborative and 10 background systems. We also compute F_u, the number of unique relevant documents as a fraction of the number of relevant documents found by a system. We repeated the analysis using the merged run results.

The 2007 TRECVID competition for *ad hoc* search consisted of 24 topics (0197-0220); the number of relevant shots varied greatly among the topics ($range = [6, 1150]$, $\bar{x} = 196$, $median = 130$, $\sigma = 234$ for the original TRECVID data, $range = [6, 1170]$, $\bar{x} = 198$, $median = 130$, $\sigma = 238$ for the augmented ground truth). This large variability made it problematic to average performance across topics, so for each topic, we ranked the scores (N_u and F_u) of the various runs, and averaged ranks rather than the raw scores across topics. Finally, we compared the average ranks we obtained from the merged runs with the collaborative runs, as shown in the rightmost columns of Table 2.

We were also interested in the temporal profile of our sessions with respect to finding unique documents. Did people find unique relevant documents after they found documents found by other systems, or were they sprinkled throughout the session? For this analysis, we again used the other groups' data as a baseline, but rather than comparing our complete runs, we used subsets of our data collected through 3.25, 7.5 and 11.5 minutes of a 15 minute session. We compared our fractional data to the full data of other systems because we were not interested in comparing directly against other systems, but in comparing between the collaborative and merged systems. The baseline data served as background, as a relatively unbiased estimate of easy to discover documents that were relevant to each topic. Given that baseline, how do our two systems compare when finding new unique documents?

As before, we performed the analysis on all topics, and then by topic size (plentiful/sparse). The results are summarized in Table 2. The data for overall uniqueness show a slight advantage for collaborative search at the end of the 15 minute session (a rank improvement of 2-3%), but that is a small gain. When we split the topics by sparsity, however, a different pictures emerges. For plentiful topics, there is no effective gain for the collaborative system (a rank improvement of 0-2%), whereas for the sparse topics there is a 12-14% uniqueness rank improvement.

Over time, there appears to be no strong trend for either metric either overall or for plentiful topics, whereas for sparse topics, improvements accrue quickly, and then reduce a bit at the end. This suggests that collaborative search teams find more unique relevant documents than merged results from individual searchers, that the advantage is greater for difficult topics, and that for those topics, there is more advantage earlier in the search session than later.

As with recall and precision measures discussed in the previous section, these results suggest that mediated collaboration is more effective when the search topics are sparse. While a more principled exploration of the effect of topic difficulty on system performance is warranted, the trend is encouraging.

	3.75 minutes			7.5 minutes			11.25 minutes			15 minutes		
	Merge	Coll	%Chg	Merge	Coll	%Chg	Merge	Coll	%Chg	Merge	Coll	%Chg
Average Rank of N_u												
Overall	7.61	7.00	+8.02	6.90	6.09	+11.7	5.88	5.74	+2.38	5.54	5.35	+3.43
Plentiful	8.92	8.83	+1.01	8.33	8.33	0.00	7.83	8.08	-3.19	7.67	7.50	+2.22
Sparse	5.00	4.56	+8.80	4.50	3.64	+19.1	3.92	3.18	+18.9	3.42	3.00	+12.28
Average Rank of F_u												
Overall	5.94	6.48	-9.09	5.60	5.26	+6.07	4.83	5.18	-7.25	5.08	4.96	+2.36
Plentiful	6.42	8.08	-25.9	6.33	6.92	-9.32	5.92	7.17	-21.1	6.75	6.75	0.00
Sparse	5.00	4.33	+13.4	4.50	3.45	+23.3	3.75	3.00	+20.0	3.42	2.92	+14.6

Table 2: Average ranks for uniqueness measures. Smaller scores represent better performance. The "%Chg" column represents the percent improvement (decrease in average rank) of collaborative (Coll) over merged (Merge).

5 Future Directions

The work described in this paper represents some initial steps in exploring the design space of algorithmically-mediated information retrieval collaboration. While our initial results are encouraging, much remains to be explored. We are pursuing several broad directions, including understanding the range of roles and the sorts of algorithms and interfaces that support them. Roles are a critical part of our system: they inform the design of user interfaces, and determine the algorithms that the regulator and back end use to retrieve and represent search results. Our initial implementation focused on the roles of Prospector and Miner, but other roles may also be useful. We are also looking at the relative contributions of team members in terms of roles and how best to allocate scarce human resources.

Our experiments were conducted on TRECVID data. There are many other possible types of data and scenarios in which to implement algorithmic mediation in the collaborative information seeking process, including but not limited to the text, web and legal/patent domains. Due to the multimodal nature of video data, our initial collaboration algorithms therefore focused on a method that could handle queries and results across this wide variety of data types: results list fusion. Future work will explore possibilities that arise by restricting the search to a single data type, i.e. text-only or image-only. In those narrower situations it should be possible to create collaboration algorithms based on the intrinsic content of the information being queried and retrieved. For example, instead of the Miner digging through the best unseen documents that result from a Prospector's query stream, the Miner might instead see a continually-updating variety of semantic facets. The Prospector might not even be aware that a certain untapped, potentially relevant facet is accruing in his or her retrieval activities. But a collaborative algorithm might analyze the unseen content and surface those facets to the Miner, synchronously.

6 Conclusion

In this paper, we have reviewed our video retrieval system, MediaMagic, and presented a novel extension of the system to incorporate realtime algorithmically mediated collaboration. In our latest, extensive evaluations of the system we found that our retrieval performance in the absence of searchable text transcripts nearly equals that of the complete system using text queries and transcripts. We also demonstrated that this implementation of mediated collaboration improved selected precision, selected recall, viewed precision, and the number of unique relevant documents found compared with naive merging of search results obtained independently by two searchers.

The algorithmic mediation in the collaborative system demonstrates the efficacy of one possible instantiation of a more general concept. Numerous challenges remain, including designing and comparing different real-time collaboration algorithms, defining additional roles, better understanding the tradeoffs between parallel and synchronized work, and designing appropriate user interfaces. Overall, we are confident that these first steps will lead to a fruitful research field, success in which will rely on the combined efforts of IR and HCI researchers.

Looking beyond video search, the notion of algorithmically-mediated explicit collaboration is a novel area for many kinds of Information Retrieval systems, from text search to web search, enterprise search and others. Within such a framework, a user is interacting not only with a human partner, but with a search engine that is also interacting with the same partner, algorithmically taking into account that partner's actions to fulfill a shared information need. These streams of information (computer retrieval plus partner search activity) are combined algorithmically in real time; they alter and influence each other. How these intertwined streams are presented to each partner, and what effect that has on retrieval effectiveness of the system, is a challenging and fruitful open question, one with considerable practical and research value.

References

[1] J. Adcock, M. Cooper, A. Girgensohn, and L. Wilcox. Interactive video search using multilevel indexing. In *CIVR 2005*, pages 205–214, 2005.

[2] J. Adcock, A. Girgensohn, M. Cooper, T. Liu, L. Wilcox, and E. Rieffel. Fxpal experiments for trecvid 2004. In *Proceedings of the TREC Video Retrieval Evaluation (TRECVID)*, pages 70–81, Washington D.C., 2004. NIST.

[3] J. Adcock, A. Girgensohn, M. Cooper, and L. Wilcox. Interactive video search using multilevel indexing. In *International Conference on Image and Video Retrieval*, pages 205–214, 2005.

[4] J. Adcock, J. Pickens, M. Cooper, F. Chen, and P. Qvarfordt. Fxpal interactive search experiments for trecvid 2007. In *Proceedings of the TREC Video Retrieval Evaluation (TRECVID)*, Washington D.C., 2007. NIST.

[5] J. A. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch and the efficient evaluation of retrieval systems via the hedge algorithm. In *Proc. SIGIR 2003*, pages 393–394, July 2003.

[6] R. Baeza-Yates and J. A. Pino. A first step to formally evaluate collaborative work. In *GROUP '97: Proc. ACM SIGGROUP Conference on Supporting Group Work*, pages 56–60, New York, NY, USA, 1997.

[7] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, 2006.

[8] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4):573–595, 1995.

[9] L. Bottou and Y. Bengio. Convergence properties of the K-means algorithms. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 585–592. The MIT Press, 1995.

[10] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo. Large-scale multimodal semantic concept detection for consumer video. In *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 255–264, New York, NY, USA, 2007. ACM.

[11] M. G. Christel and R. Yan. Merging storyboard strategies and automatic retrieval for improving interactive video search. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 486–493, New York, NY, USA, 2007. ACM.

[12] M. Cooper, J. Adcock, and F. Chen. Fxpal at trecvid 2006. In *Proceedings of the TREC Video Retrieval Evaluation (TRECVID)*, 2006.

[13] M. Cooper, J. Adcock, H. Zhou, and R. Chen. Fxpal at trecvid 2005. In *Proceedings of the TREC Video Retrieval Evaluation (TRECVID)*, 2005.

[14] A. Girgensohn, J. Adcock, M. Cooper, and L. Wilcox. A synergistic approach to efficient interactive video retrieval. In *INTERACT*, pages 781–794, 2005.

[15] G. Golovchinsky. What the query told the link: the integration of hypertext and information retrieval. In *HYPERTEXT 1997*, pages 67–74, New York, NY, USA, 1997. ACM.

[16] P. I. Good. *Permutation, Parametric, and Bootstrap Tests of Hypotheses (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2004.

[17] A. Hauptmann, W.-H. Lin, R. Yan, J. Yang, and M.-Y. Chen. Extreme video retrieval: Joint maximization of human and computer performance. In *Proc. ACM Multimedia 2006*, pages 385–394, Santa Barbara, CA, 2006.

[18] A. G. Hauptmann, W.-H. Lin, R. Yan, J. Yang, and M.-Y. Chen. Extreme video retrieval: joint maximization of human and computer performance. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 385–394, New York, NY, USA, 2006. ACM.

[19] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 762, Washington, DC, USA, 1997. IEEE Computer Society.

[20] M. A. H. Huijbregts, R. J. F. Ordelman, and F. M. G. de Jong. Annotation of heterogeneous multimedia content using automatic speech recognition. In *Proceedings of the Second International Conference on Semantic and Digital Media Technologies, SAMT 2007, Genoa, Italy*, volume 4816 of *Lecture Notes in Computer Science*, pages 78–90. Springer Verlag, 2007.

[21] X. Li, D. Wang, J. Li, and B. Zhang. Video search in concept subspace: a text-like paradigm. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 603–610, New York, NY, USA, 2007. ACM.

[22] H.-B. Luan, S.-X. Lin, S. Tang, S.-Y. Neo, and T.-S. Chua. Interactive spatio-temporal visual map model for web video retrieval. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 560–563, 2–5 July 2007.

[23] Lucene. Jakarta lucene. http://jakarta.apache.org/lucene/docs/index.html.

[24] M. R. Morris and E. Horvitz. Searchtogether: an interface for collaborative web search. In *Proceedings of UIST*, pages 3–12, 2007.

[25] M. R. Naphade, L. Kennedy, J. R. Kender, S.-F. Chang, J. Smith, P.Over, and A.Hauptmann. A light scale concept ontology for multimedia understanding for trecvid 2005. Computer Science Technical Report RC23612 W0505-104, IBM, 2005.

[26] P. Over, G. Awad, W. Kraiij, and A. Smeaton. Trecvid 2007 an overview. In *Proceedings of the TRECVID 2007 Workshop*, Nov. 2007.

[27] C. Petersohn. Fraunhofer hhi at trecvid 2004: Shot boundary detection system. In *Proceedings of the TREC Video Retrieval Evaluation (TRECVID)*, pages 64–69, Washington D.C., 2004. NIST.

[28] A. F. Smeaton and P. Browne. A usage study of retrieval modalities for video shot retrieval. *Inf. Process. Manage.*, 42(5):1330–1344, 2006.

[29] A. F. Smeaton, H. Lee, C. Foley, S. McGivney, and C. Gurrin. Físchlár-diamondtouch: Collaborative video searching on a table. In *Multimedia Content Analysis, Management, and Retrieval*, San Jose, CA, January 15-19 2006.

[30] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 321–330, New York, NY, USA, 2006. ACM.

[31] B. Smyth, E. Balfe, O. Boydell, K. Bradley, P. Briggs, M. Coyle, and J. Freyne. A live-user evaluation of collaborative web search. In *Proc. IJCAI 2005)*, pages 1419–1424, Ediburgh, Scotland, 2005.

[32] C. Snoek, M. Worring, D. Koelma, and A. Smeulders. A learned lexicon-driven paradigm for interactive video retrieval. *IEEE Trans. on Multimedia*, 9(2):280–292, Feb. 2007.

[33] C. G. M. Snoek, I. Everts, J. C. van Gemert, J.-M. Geusebroek, B. Huurnink, D. C. Koelma, M. van Liempt, O. de Rooij, K. E. A. van de Sande, A. W. M. Smeulders, J. R. R. Uijlings, and M. Worring. The mediamill trecvid 2007 semantic video search engine. In *Proceedings of the 5th TRECVID Workshop*, November 2007.

[34] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430, New York, NY, USA, 2006. ACM.

[35] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1088–1099, 2006.

[36] M. B. Twidale, D. M. Nichols, and C. D. Paice. Browsing is a collaborative process. *Information Processing and Management*, 33(6):761–783, 1997.