Experiments in Interactive Video Search by Addition and Subtraction

John Adcock
FX Palo Alto Laboratory
3400 Hillview Ave, Bldg. 4
Palo Alto, California 94304
adcock@fxpal.com

Matthew Cooper
FX Palo Alto Laboratory
3400 Hillview Ave, Bldg. 4
Palo Alto, California 94304
cooper@fxpal.com

Jeremy Pickens
FX Palo Alto Laboratory
3400 Hillview Ave, Bldg. 4
Palo Alto, California 94304
jeremy@fxpal.com

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. This is accomplished by providing a hierarchical segmentation, streamlined interface, and redundant visual cues throughout. The typical search scenario includes a single searcher with the ability to search with text and content-based queries. In this paper, we evaluate new variations on our basic search system. In particular we test the system using only visual content-based search capabilities, and using paired searchers in a realtime collaboration. We present analysis and conclusions from these experiments.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval: Information Search and Retrieval**]: search process, retrieval models

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

Multimedia information retrieval, interactive video search, collaborative information retrieval

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, search technology within such collections in the absence of manually created indices remains relatively primitive. 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.

Figure 1: Two collaborating searchers in action

Over the last several years our group has developed a video search interface, named MediaMagic [1, 11, 10], designed to enable users to efficiently assess search results using a flexible interface and rich visualizations. Searches can be performed at the textual, visual, and semantic level. Results are displayed in 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. Meanwhile keyboard shortcuts are available for nearly all actions to reduce the amount of mousing required to sift through results lists and increase the searcher's engagement with the search task.

The video information retrieval problem is the focus of a growing research community which includes the TRECVID evaluations [21, 25]. We have participated in TRECVID since 2004 and its protocol forms the experimental framework 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 [8, 9, 13, 16, 17, 23, 27, 26].

In this paper, we present two variations of our interactive search system. First, we present a version of our search interface that makes no use of text information and operates

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIVR'08, July 7–9, 2008, Niagara Falls, Ontario, Canada.
Copyright 2008 ACM 978-1-60558-070-8/08/07 ...$5.00.

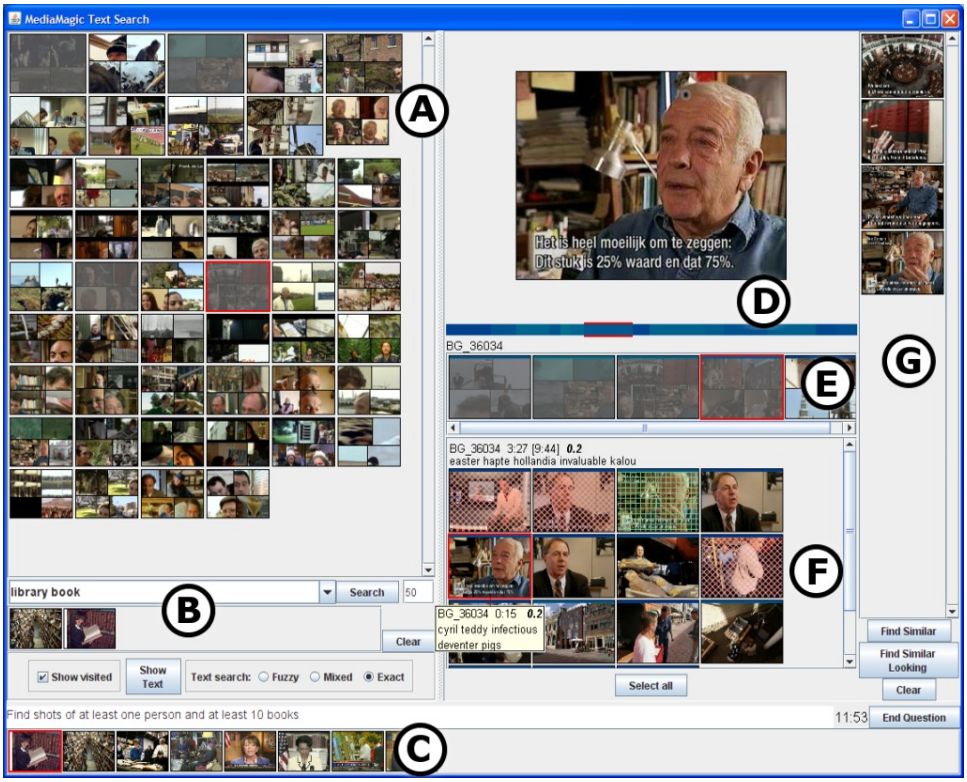

Figure 2: Interactive system interface. (A) Search results area with story keyframe summaries. (B) Search text and image entry. (C) Search topic and example media. (D) Media player and keyframe zoom. (E) Story timeline. (F) Shot keyframes. (G) Relevant shot list.

solely on the visual content of the video. We describe various required changes to our data pre-processing and interface and present experimental results demonstrating competitive performance with our standard system which includes text transcript search and indexing. Second, we present a multi-user variant of our interface, and report improved performance relative to our standard single user system.

2. MEDIAMAGIC

Our baseline search system, MediaMagic, comprises analysis, interactive operation, and post-interactive components. Our analysis begins with data pre-processing to generate indices into the 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 final search results. Much of the system has been documented elsewhere[1, 11, 10, 2]; we include a brief overview here for completeness.

2.1 Data pre-processing

2.1.1 Segmentation

The bootstrapping data-processing step is 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 [2]. 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 [18] (for keyword search) and our latent semantic indexing system (for fuzzy text search). For the latent space, we build a latent space [5] of fixed dimension 100 using the sparse vectors of stopped and stemmed terms.

2.1.3 Visual Indexing

Color correlograms [14] 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 the semantic indexing described in the following section. 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-squared distance between each frame histogram and the centroid for the the shot. Finally, SURF descriptors with their vertical pixel (Y-axis) location are also computed [4] and quantized into 200 bins using online k-means [6]. The quantized SURF descriptors are used together with the keyframe's color histogram features for semantic concept detection.

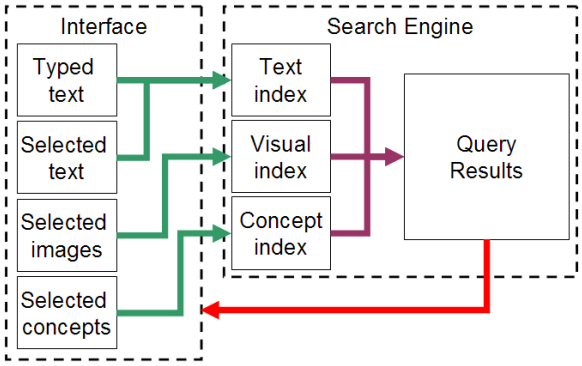

Figure 3: Search engine overview

2.1.4 *Semantic Indexing*

We use SVM-based concept detectors for the 35 lscom-lite concepts [20] to provide semantic similarity measurements. We construct single concept detectors for the lscom-lite concept set using support vector machines (SVMs). We use reduced training sets for parameter tuning and train our detectors using asymmetric bagging [29]. For each concept we generate a separate training set by randomly downsampling the set of training examples. Denote the positive and negative training examples used for classifier construction by $\mathcal{T}^+$ and $\mathcal{T}^-$, respectively. Then

$$\begin{aligned} |\mathcal{T}^+| &= \min(990, |\{\text{all positive samples}\}|) \\ |\mathcal{T}^-| &= \min(1800, 9 \times |\mathcal{T}^+|) \quad . \end{aligned}$$

The choices for these training set sizes were not systematically optimized but produced good performance. We take an "early fusion" approach, concatenating the color histograms and quantized SURF data to represent each keyframe.

Given the reduced training set $\mathcal{T} = \mathcal{T}^+ \cup \mathcal{T}^-$, we perform a basic parameter optimization via grid search using LibSVM [7]. Specifically, we learn C,which is the penalty for misclassifications, and γ, which scales the radial basis kernel function used by the SVM. We then train three separate SVMs using the learned parameters. For each we use different training sets by resampling the development data using the proportions of positive and negative examples described above. 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 [28].

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 interactive search system 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 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 [18] 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 [5], 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. A query returns a ranked list of stories.

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 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. When text is not available (see section 2.3), only the concept similarity is used.

2.2.4 Visual cues

The search engine performs retrieval of story segments based on the indices described above and depicted in Figure 3. 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 excluded from the relevant shot set (see Figure 2 F). A green overlay on a shot icon indicates that it has been included in the results set (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 2.1.4) for each shot instead of the MT/ASR transcript. We use the same novelty-based segmentation of [2] 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).

The analysis tools we leverage here have been used in our system previously. Their use in search without text information is more common in non-interactive search systems (e.g. automatic search systems). Increasingly, groups are shifting the emphasis of interactive systems away from complex content analysis and reliance on text towards interfaces which present users with a large number of search results and a variety of tools for subsequent exploration and query refinement. The MediaMill cross and fork browsers [27] and CMU Informedia's *Extreme Video Search* [13] browsers also exemplify this trend. These are our first experiments in which text is altogether absent from both the indexing and search processes.

3. MULTIPLE USER COLLABORATION

The second direction in which we have extended our basic search system enables multiple users to perform collaborative search. The terms "collaboration" or "collaborative search" are overloaded with many meanings, ranging from multiple searchers working separately in parallel, with shared interface awareness [19], multiple users sharing a single interface [24], to the "Web 2.0" sense of collaborative filtering or personalization. The goal of our system is to support the explicit collaboration of focused search teams whose activities are mediated algorithmically. The information that one member finds is used by the underlying system to automatically influence the otherwise independent search behaviors of the team members. Explicit collaboration is conceptually distinct from crowd-based collaboration algorithms and interfaces that support re-finding and re-discovering of already-found information. Instead, our explicit collaboration model provides algorithms and interfaces that support exploration and the discovery of relevant information that has not yet been found.

We extend the MediaMagic client with 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.

3.1 System Architecture

The architecture for our collaborative system is generic and can be adapted to modalities other than video. It consists of three parts: the User Layer, the Regulator Layer, and the Algorithmic Layer (see Figure 4).

3.1.1 User Layer

The user layer contains all the input and output devices for human-computer interaction. Within the user layer is the MediaMagic video search interface for issuing image, text and concept queries and browsing results. A rapid serial

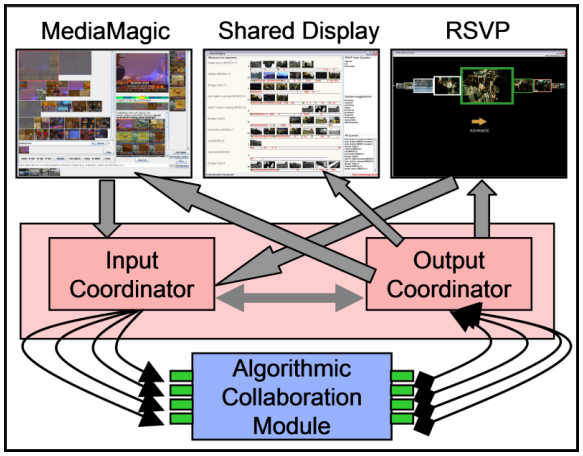

Figure 4: The Collaborative System Architecture

visual presentation (RSVP) interface is used for rapid display and relevance assessment of video shots. The RSVP interface is similar in motivation and application to CMU Informedia's *Extreme Video Search* [13], and bears some resemblance to existing work on RSVP interfaces for video [30]. Finally, a shared display shows the collaborated search activities of the two users, along with information derived from their activities, such as system suggested queries.

3.1.2 Regulator Layer

Within the regulator layer there is an input regulator and an output regulator. The input regulator is responsible for intercepting searcher activities, such as queries and relevance judgments, and contains coordination rules that then call the appropriate subset of algorithmic collaboration functions. In effect, the input regulator enforces a policy that allows the users to act in certain predetermined collaborative roles. The output regulator is similar, accepting information from the algorithmic layer and routing the correct information to the appropriate user or information display at the appropriate time.

3.1.3 Algorithmic Layer

The algorithmic layer consists of functions for combining the activities of two or more searchers to produce documents, rankings, query suggestions, or other pieces of information relevant to the search. The objective of this entire algorithmically mediated collaborative search architecture is to ensure that the best information flows seamlessly to the right searcher at the right time, so that they can be the most effective in completing their search task. This should happen with minimal extra effort from the other collaborators.

3.2 Collaborative Search Roles

The synchronous and explicit nature of the collaboration enables searcher specialization. In our system, collaborating users adopt the specialized, complementary roles of *Prospector* and *Miner*, supported respectively by the MediaMagic and RSVP interfaces, and the underlying algorithms connecting them. The role of *Prospector* is designed to allow a user to open up new avenues of exploration into the collection, while the role of *Miner* insures that richer veins of information are more quickly and effectively explored.

Figure 5: Shared Display interface (top) and RSVP interface (bottom)

When a MediaMagic user enters a query, the input regulator issues that query against a search engine, stores the list of results, and routes the list back to both the MediaMagic client and the shared display via the output regulator. When the MediaMagic client enters more queries the same thing happens. Any shots that either the MediaMagic client or the RSVP client subsequently mark relevant are passed to the input coordinator as well and stored.

When the RSVP client makes a request for a set of shots (30 at a time) the input regulator responds. First, it filters shots the MediaMagic user has already examined to avoid duplication of effort. It calls the collaborative algorithmic layer to get the best set of shots that have been retrieved by the MediaMagic queries, but have not yet received any search team attention. Thus, the RSVP client does not manually select shots to comb through, reducing the cognitive load. Moreover, the shots that get fed to the RSVP user change constantly depending on what the MediaMagic user is doing; searcher activities are algorithmically coordinated.

The third major part of the interface is the shared display, a large screen in the front of the room in Figure 1. It shows continually updating data about the current search run: the queries that have been issued, the relative ranks of the shots that were retrieved by those queries, and the associated relevance, non-relevance, or unseen state of each of those queries. It also scrolls through visual thumbnails of the relevant shots that have been retrieved by a particular query. Most important to the overall collaboration is the area showing collaborative system-suggested query terms. It

is through this interface that the activities of the RSVP user are fed back to the MediaMagic user. This will be described in the next section.

3.3 Collaborative Algorithm

The main purpose of the collaborative algorithm is to alter, in real time, the information presented to each search team member, based on the activities of the search team as a whole. There are two parts to this algorithm. The first part is how the shots fed to the RSVP user are chosen. The second part is how the system-suggested query terms are fed, via the Shared Display, to the MediaMagic user.

3.3.1 RSVP Shot Priority

This algorithm determines the order in which unseen shots are fed to the RSVP user. The foundation of the algorithm is weighted Borda count [3] fusion. When a query is issued, the higher in that ranked list an unseen shot appears (Borda count), the greater its position in the priority queue of shots to send to the RSVP user. However, this rank information is tempered by the overall quality of the query to which a shot belongs. Two weighting factors are used, query freshness (w_f) and query relevance (w_r).

Query freshness is given by the ratio of unseen to seen results that have been retrieved by that query: $w_f = \frac{unseen}{seen}$. Query relevance is given by the ratio of relevant to non-relevant shots that have been found in the seen results for a query: $w_r = \frac{rel}{nonrel}$. These two factors counterbalance each other. If a query has been successful in retrieving a lot of relevant shots, you want the RSVP user to continue examining those shots (relevance). However, if most of the shots from that query have already been examined, you want to give other queries priority (freshness). Similarly, a query with only a few examined shots receives high priority (freshness). However, after a few sets of shots have been examined and turn out not to be relevant, remaining shots from that query are downplayed (relevance). Underscoring both the relevance and freshness weights is the original Borda count (rank) given to the shot. If a shot is found in more than one query queue, its weighted value is summed.

3.3.2 System-Suggested Term Selection

While the actions of the MediaMagic user (queries performed, shots examined, shots marked relevant) have an effect on the ordering of the shots fed to the RSVP user, the actions of the RSVP user (shots examined, shots marked relevant) drive the system-suggested query terms fed to the MediaMagic user. The basic idea is similar to above, with relevance and freshness weights. However, instead of a Borda count, a "term frequency in the query" (tf_q) count is used. When a query is issued, all the term counts associated with all the separate shots retrieved by that query are summed. The higher the frequency of that term in the retrieved set, the higher its priority for appearing as a system-suggested term. However, this tf_q count is tempered by the same two factors: relevance and freshness. The more relevant documents are found in a query, the higher the weight on that count. As more shots in that query are examined, that query loses freshness, and tf_q counts are down-weighted. In this manner, the RSVP user's activity constantly and automatically updates the system-suggested term list. The more the RSVP user explores fresh, relevant pathways, the more the associated terms related to those pathways appear. No cognitive load is required for the RSVP user to suggest terms, just as no cognitive load is required for the MediaMagic user to tweak the order of shots fed to the RSVP user.

Name	MAP	Description
CO15	0.247	15 min. COllaborative search
SUA	0.214	15 min. Single User search with All information (including text) used
CO11	0.211	COllaborative search with simulated stop at 11.25 min.
SUV	0.208	15 min. Single User search with only Visual information (no text) used
CO7	0.162	COllaborative search with simulated stop at 7.5 minutes
SUA_b	0.149	Supplementary 15 min. Single User search with All information (including text) available

Table 1: MAP scores in decreasing order for the search systems from our TRECVID 2007 evaluations. Underlining is used in the descriptions to clarify the source of our system abbreviations.

4. EXPERIMENTS

In this section we review experiments conducted in the context of the TRECVID 2007 evaluation [21]. 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 [15] and machine-translations to English (referred to below as MT/ASR), were provided, along with a common shot-level segmentation [22]. 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.

4.1 Summary Performance

Mean average precision (MAP) is the principal summary performance metric used in the TRECVID evaluations. Table 1 shows descriptions of the system variants along with their MAP scores. 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, but 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 other systems (CO15,CO11,CO7) employ the real-time, multi-user, collaborative search system described in section 3 and include text search.

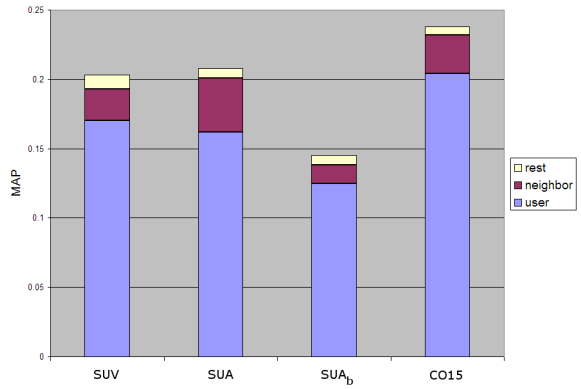

Figure 6: MAP on the 24 TRECVID 2007 topics broken down by contribution from user-selected shots (user), then adding neighboring shots (neighbor), and then the rest of the submitted shots (rest). Note that the visual-only run (SUV) outperforms the visual+text run (SUA) when only user-identified shots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

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 final post-interactive query list. The performance is dominated by the shots identified directly by the user.

We evaluated the statistical significance of the measured MAP differences using a standard permutation test [12]. The test measures the chance that a random assignment of average precision scores between the two systems yields the same or greater difference in MAP. We find within the collaborative variations that the measured differences CO15 > CO11 > CO7 are all significant at a level greater than p=0.01. This result is reassuring since the CO11 and CO7 systems are simply time-truncated versions of CO15. If this were not the case it would imply that the user had no systematic contribution to the results after the first 7.5 minutes of search time. Also unsurprising given the very small measured difference between the with and without-text single-user systems, SUA and SUV, is that this observed difference is not statistically significant.

After these results, the most consistent difference is that SUA$_b$ performs reliably worse (significant at p=0.02 or better) than every other system except the 7.5 minute collaborative system, CO7. 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 outperforms the single-user systems at 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.

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.

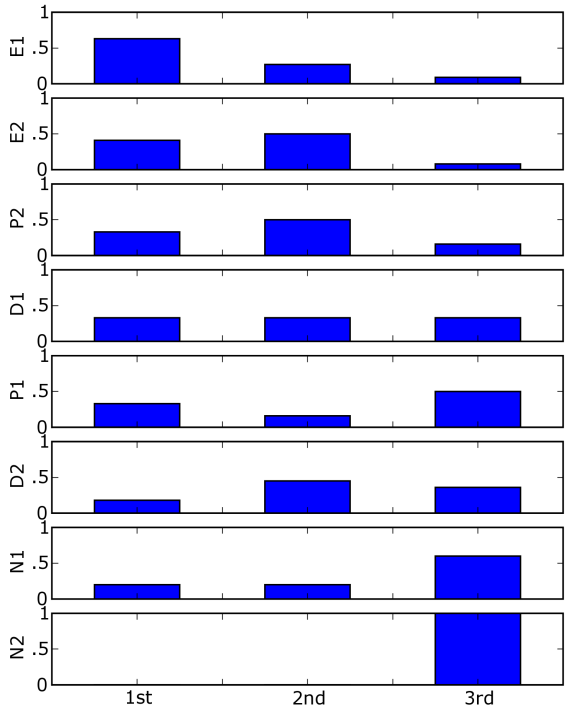

Figure 7: Normalized histograms of searcher average precision ranks across all topics for the 3 standalone runs (SUA, SUV, and SUA$_b$) sorted by Borda score. In decreasing order of familiarity with the system are: (E)xperts, (D)evelopers, (P)ros, and (N)ovices.

4.2 Searcher Performance

For each of the 24 seach 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 continuity 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 Comparative Analysis

There are two ways of looking at the performance improvements of the collaborative runs relative to the single user runs. First, the CO15 run shows a 14.6% MAP improvement over SUA, and a 16.9% improvement over SUV. A second way of measuring improvement is to calculate the amount of time it takes to obtain an equivalent MAP score to SUA. Our CO11 run was simulated by submitting all the results obtained by the collaborative team at 11 minutes and 15 seconds. Table 1 shows that the same MAP can be obtained in 75% of the time, a 25% improvement.

While the above metrics show improvements, they provide only limited insight into the relative performance differences of the systems. For deeper analysis, we first examine the results on a per-topic basis. We also study actions performed during the actual runs (how many relevant shots were found, how many non-relevant shots were examined, etc.), rather than post hoc padded results. Finally, we will examine variation of these measures with the true number of relevant shots[1]. In the following subsections we will compare CO15 with SUA and an artificial collaborative run: SUA+V, or "merged". Given that the SUA and SUV systems performed at roughly the same level, we wondered how well the post hoc combination these two runs would perform. Duplicate relevant shots are removed, and duplicate non-relevant shots are removed as well, simulating the effect of an interface-only collaborative system in which users are simply made aware of the previous search activities of their partners, but no algorithmic support is made available.

4.3.1 Recall and Precision

Here we analyze the precision and recall of the shots actually viewed by users during the 15 minute runs, rather than padded ranked lists from the end of a run which include automatically selected shots as detailed in section 2.2.5. The breakdown within the submitted results per run is detailed in Figure 6. We first define notation. Denote the set of relevant shots by $\mathcal{R}$ (pooled across all TRECVID 2007 participants) for a given topic. Denote the set of shots labeled as relevant or positive in a submitted run by $\mathcal{P}$. Recall is

$$\text{Recall} = \frac{|\mathcal{P} \cap \mathcal{R}|}{|\mathcal{R}|} \quad .$$

[1]This is approximated using the pooled submissions for TRECVID. See [21] for more details.

Precision is

$$\text{Precision} = \frac{|\mathcal{P} \cap \mathcal{R}|}{|\mathcal{P}|} \quad .$$

For precision, CO15 shows a per-topic average 1.5% improvement over SUA and a 15.4% improvement over SUA+V. These differences are not statistically significant. We frankly did not expect them to be, because precision alone, in the manually-retrieved set, essentially measures user agreement with NIST judgments. We would not expect any system in which the same set of users are making an honest effort to find relevant shots to differ significantly in precision alone. For recall, CO15 shows a per-topic average 101.1% change compared to SUA, 43.7% change compared to SUV, and −10.7% change compared to SUA+V. Using a permutation test as in section 4.1, these results were significant at levels of p=0.01, p=0.09, and p=0.03 respectively. Collaborative retrieval outperforms the standalone systems, but as we will see in section 4.3.3 some of those improvements are only slight, while some are large.

It appears that these metrics confirm what we learned in section 4.1: collaborative search outperforms single user search. Unfortunately, it does not appear that, on average, it outperforms the combined standalone runs, SUA+V. For the collaborative system, precision is slightly better and recall is slightly worse. In the next sections we examine factors that are obscured within these average measures. The first is the number of shots that the user(s) of a system manually review. The second is the size of the topic.

4.3.2 Normalized Recall and Precision

We wish to understand whether one system is performing better than another simply because it has been able to put more shots in front of the user, or whether there is an underlying systematic improvement. We can explore this with per-shot statistics. To normalize recall and precision for each topic, we take the the count of manually identified relevant shots and divide it by the the count of shots examined during the course of that topic's run. We denote the set of examined (seen) shots for a given topic and system by $\mathcal{S}$. This gives us an effort-normalized value of the relevant shot count. The idea is to compensate for brute force approaches. The metric for precision does not change:

$$\text{Normalized Precision} = \frac{\frac{|\mathcal{P} \cap \mathcal{R}|}{|\mathcal{S}|}}{\frac{|\mathcal{P}|}{|\mathcal{S}|}} = \frac{|\mathcal{P} \cap \mathcal{R}|}{|\mathcal{P}|} \quad .$$

Recall, on the other hand, changes to

$$\text{Normalized Recall} = \frac{\frac{|\mathcal{P} \cap \mathcal{R}|}{|\mathcal{S}|}}{|\mathcal{R}|} = \frac{|\mathcal{P} \cap \mathcal{R}|}{|\mathcal{R}||\mathcal{S}|} \quad .$$

The average (across topics) unique (non-duplicated) examined shot counts of the various system are as follows: SUA = 2123, SUV = 2601, SUA+V (merged) = 4184, CO15 = 2614. When normalizing by these values (using the actual values for each topic, rather than the these averages), CO15 does 73.9% better than SUA and 44.1% better than SUA+V.

This shows that the collaborative system is actually more effective than not only the standalone system, but the merged system as well. SUA gets through fewer shots than CO15, so there is a lower normalization factor for SUA, which increases its overall score. (In the unnormalized version, CO15 had 101.1% higher recall; normalization lowers this to

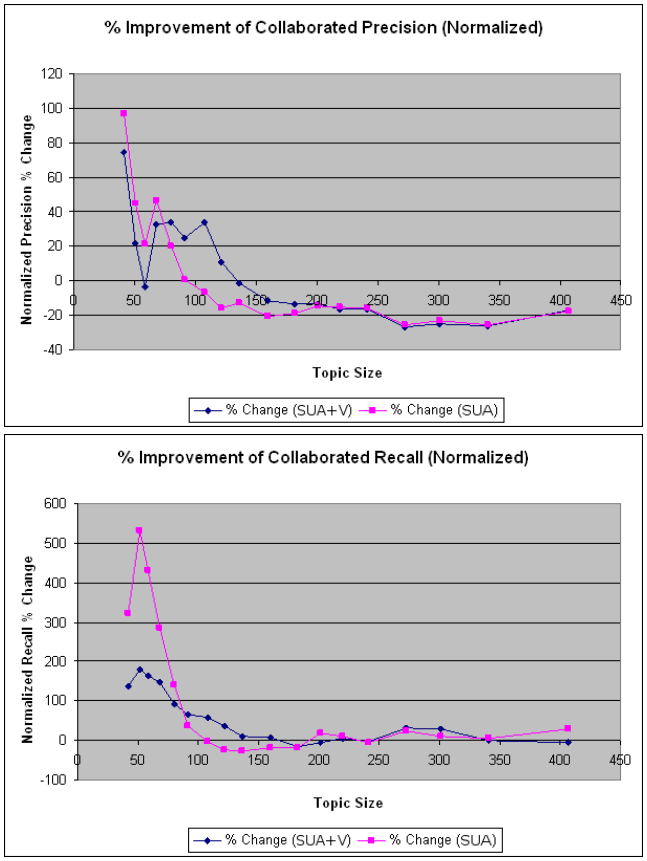

Figure 8: Precision (top) and Recall (bottom) performance metrics plotted as a function of topic size.

73.9%). Even still, C015 outperforms SUA by a significant margin. Similarly, SUA+V (merged) goes through more shots than CO15, but there is also more thrashing: more shots are required to obtain similar recall numbers. It was a surprising disadvantage of our system that users did not review more shots than the single-user variants. This was due mainly to an RSVP interface that was not properly optimized. Nevertheless, these results show that the collaborative system was able to work "smarter" than either of the other systems.

4.3.3 *Per Topic Normalized Recall and Precision*

We also found that topic size (number of available relevant shots) is an important factor for comparing our systems. Figure 8 contains two graphs, one for recall and one for precision. Along the x-axis are the various topics, represented by the total number of relevant documents available in the collection (from the NIST ground truth). Along the y-axis are the percentage differences between CO15 and either SUA or SUA+V. In order to get a better sense of the patterns inherent in this data, values have been smoothed using kernel regression with simple exponentially decaying windows. Notice that the improvements are not randomly distributed. Where CO15 most outperforms SUA and SUA+V is on topics with fewer total available relevant shots, haystacks with fewer needles. The CO15 performance on these more "difficult" topics is *much* better.

5. CONCLUSION

In this paper, we have reviewed our basic video retrieval system, MediaMagic, and presented two novel extensions. The basic system emphasizes rich visualization of retrieval results computed using simple yet powerful automatic analysis. The first extension of our system is a text-free version which replaces textual similarity with semantic similarity. The semantic similarity is determined using output probabilities of automatic semantic concept detectors. Performance of this system in a large scale evaluation very nearly equals that of the basic system using text queries and transcripts.

We also presented a multi-user collaborative variant of the system which integrates an RSVP interface into the interaction. The collaborative system uses algorithmic mediation and a shared display to coordinate the activities of the two searchers. We reviewed extensive analysis comparing the performance of the collaborative system to that of our single user system, and a simulated "merged" run combining the results from two independent searchers. Our preliminary analysis reveals an interesting tradeoff for collaborative search . When there are relatively many relevant shots to be found, it appears that searchers should be freer to work on their own, without algorithmic collaboration. There are going to be enough available relevant shots that each independent searcher can spend all 15 minutes working separately. However, when relevant shots are more difficult to come by, two searchers working independently are not quite able to find them. On the one hand, this seems counter-intuitive: With more people searching for something, the chances are greater that any one person will find it. On the other hand, this demonstrates the advantages of algorithmically collaborated search; it is only when two searchers' activities are coordinated by the underlying system that they are more able to push further into the collection, and find those nuggets, than had they been working separately.

6. REFERENCES

[1] 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.

[2] 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.

[3] J. A. Aslam and M. Montague. Models for metasearch. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284, New York, NY, USA, 2001. ACM.

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

[5] 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.

[6] 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.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[8] 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.

[9] 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.

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

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] M. R. Morris. Interfaces for collaborative exploratory web search: Motivations and directions for multi-user designs. In *CHI 2007 Workshop on Exploratory Search and HCI*, 2007.

[20] 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.

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

[22] 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.

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

[24] A. F. Smeaton, H. Lee, C. Foley, S. McGivney, and C. Gurrin. Físchlár-diamondtouch: collaborative video searching on a table. In *SPIE Electronic Imaging - Multimedia Content Analysis, Management, and Retrieval*, pages 15–19, 2006.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] K. Wittenburg, C. Forlines, T. Lanning, A. Esenther, S. Harada, and T. Miyachi. Rapid serial visual presentation techniques for consumer digital video devices. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 115–124, New York, NY, USA, 2003. ACM.