Searching Visual Semantic Spaces with Concept Filters

Eric Zavesky*, Zhu Liu, David Gibbon, Behzad Shahraray
AT&T Labs Research, Middletown, NJ
{ezavesky, zliu, dcg, behzad}@research.att.com

Abstract

Semantic concepts cement the ability to correlate visual information to higher-level semantic concepts. Traditional image search leverages text associated with images, a low-level content-based matching, or a combination of the two. We propose a new system that uses 374 semantic concepts (derived from the LSCOM lexicon [6]) to semantically facilitate fast exploration of a large set of video data. This new system, when coupled with traditional image search techniques produces a very intuitive and fruitful design for targeted user interaction.

1 Introduction

The signal processing community has long studied low-level features and derived high-level features (or semantic concepts) for large image databases. High-level concepts are generally learned using patterns discovered over a set of images, where machine learning techniques are used to create discrete classifiers and provide a deterministic scores for concept similarity. At the root of high-level features, two primary approaches are commonly used. One approach analyzes low-level content similarity to find common patterns and the other, more commonly used in the computer vision field, is to try to learn explicit object descriptors, often involving its geometry or a majority of the object's views. While more exact, establishing a single appearance for an object may prove difficult or computationally infeasible. For example, the semantic concept of a tree, has an infinite number of visual appearances must be learned to have a model generic enough to be accurate. We extend our approach to a diverse set semantic concepts, so we adopt the first approach, based on low-level content similarity.

Content-based search and indexing is a diverse task that has been well studied, but an optimal general solution is still not available. Early experiments on large image databases first extracted low-level image features and attempted to heuristically search and match content over a given dataset [10]. Later, authors conjectured that brief text descriptions (often referred to as tags) could be associated with image regions and language processing techniques were introduced to couple image and text features [1]. The breadth of additional approaches is quite large, as authors experimented with local region correlation [12], multi-modal features in complex machine learning environments [2], and even explored image correlation in a concept space [5]. While each approach has had gains in its targeted area, none have provided a complete answer for searching where fast-response user interaction is required.

In this work, we leverage the scores of a large set of semantic classifiers to enable the user to quickly navigate a large, diverse image database derived from captured broadcast television. We apply previous classifier methods presented in other works and instead focus on enabling user interaction over the automatically computed concept space. The remainder of this paper is divided into three major sections: section 2 has descriptions of our system and features, section 3 details our experiments and our collected dataset, and finally section 4 contains concluding remarks and describes future work.

2 Architecture and features

Our system design capitalizes on the modular design of several image processing and data management components to maximize performance in a parallel environment. As discussed in the last section, the *Miracle* platform [3] was developed at AT&T Labs to capture, index, and search a large set of video data. This platform then allows a variety of conetnt clients to search, browse, and retrieve the indexed video data; PDA's, portable game devices, and even cell phones can navigate recorded media and playback bandwidth-appropriate media. For this work, we integrated low-level feature extraction and high-level feature classifiers to further navigate a very large video corpus. The following sections describe the *Miracle* system, features used for semantic classification, the integration steps for semantic information, and finally our user interface.

*Collaborating researcher from DVMM Lab, Columbia University, NY

2.1 Miracle

Miracle [3],[7] is an ongoing research project at AT&T Labs aimed at creating automated content-based media processing algorithms and systems to collect, organize, index, mine, and re-purpose video and multimedia information. Figure 1 illustrates the overview structure of the *Miracle* system. *Miracle* is composed of three main modules: content acquisition, content processing, and content server. The content acquisition module records selected broadcast TV programs from a variety of sources, including Digital Satellite System (DSS) receiver, Cable TV (CATV), and Digital TV (DTV) terrestrial broadcasting. The content processing module transcodes the acquired content into different formats and extracts the embedded hierarchical content structure for query and browsing purposes. The content server module performs multimedia information retrieval and provides a user friendly interface, such that the users are able to effectively search interesting content in the large media archive and pleasantly browse the retrieved media clips using preferred devices, either a standard desktop or a PDA.

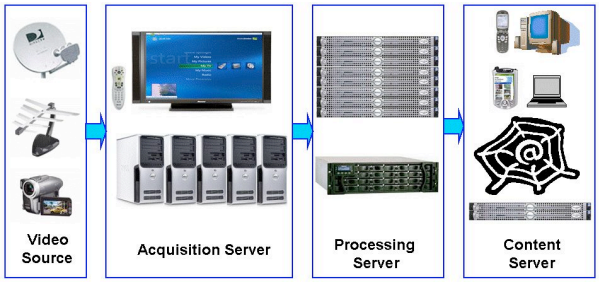

Figure 1. Architecture of the *Miracle* system.

The current *Miracle* acquisition module is built on Microsoft XP Media Center Edition (MCE) platform. MCE machine is a full fledged PVR (Personal Video Recorder) system, having the capability to schedule a single or a series of recordings of TV programs with integrated EPG support. MCE records a television show in DVR-MS file format, similar to Microsoft's ASF (Advanced Streaming Format), which allows the creation of key PVR functionalities like time-shifting, live pause, and simultaneous record and playback.

The goal of the multimedia processing is to extract a semantically meaningful index to facilitate query response and to transcode the media in different format for easy browsing. Figure 2 shows the diagram of the *Miracle* processing module. The source of the processing module is a DVR-MS file, and the processing results are stored in two databases: the media database for serving the content and the content index database for content query purpose.

The metadata in DVR-MS includes some key information about the show, for example, the program ID, the

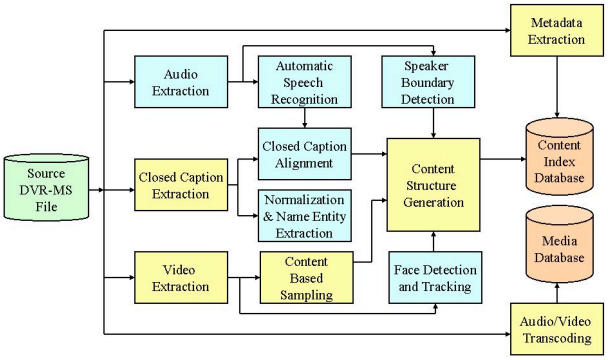

Figure 2. The *Miracle* processing module.

broadcast time, and some brief description of the content coming from the EPG; metadata is extracted and saved in the content index database. While a high quality video is usually more pleasing and may carry more information, video replay is not always an option. A different visual presentation of the video content can be done by selecting a subset of representative frames to convey the visual information. We use the algorithms discussed in [8], [9] for performing content-based sampling. This algorithm detects abrupt and gradual transitions in the video sequence, and the set of frames retained generates a compact representation of the video program.

Closed captions (CC) contain rich content information about the program, are used to search for appropriate video segments. Unfortunately, CC is normally not synchronized with the audio, which noticeably affects the quality query resutls. So, a large vocabulary automatic speech recognition (ASR) is used to generate transcripts for the audio stream. After ASR, either parallel text alignment is performed to align the timing information from the automatic speech transcription with the more accurate CC transcription or we import high quality off-line transcripts of the program when they become available. A case restoration module system uses a rule-based capitalization algorithm trained from multiple sources, including AP newswire data and online stories published by national media companies to restore case information in CC and ASR transcripts. To better index and present the content, named entities, including country names, person names, locations, titles, etc. are extracted from the textual stream.

In the content structure generation block all content index information is combined and a page/paragraph structure of the media is created. Each paragraph is composed of one scene cut frame and a set of related CC sentences. Such structure effectively represents the video data in a manner that is easy for users to browse the content nonlinearly.

Due to the wide range of accessing devices with different network and video rendering capabilities, the

DVR-MS files are transcoded to three Windows Media Video (WMV) formats: standard definition (SD) video (2Mbps/640x480/29.97fps), VHS-quality video (600Kbps/320x240/29.97fps), and low bandwidth (LB) video (150Kbps/224x168/15fps). On a modern desktop PC, the user can enjoy the standard definition video, and a PDA user can smoothly playback the low bandwidth video.

To motivate research and development, we implemented and continue to maintain a fully functioning prototype video search engine. Figure 3 shows a results page for a user query for "life on mars". In addition to document metadata, multimedia paragraphs are displayed with closed caption text; images and text link directly to the video or other media (as per the user preference), or the user may select "full program" to browse through a particular program. User may execute the query on different databases including speech or transcription data. The *Miracle* search engine currently operates on an archive of more than 41,000 discrete video broadcasts that have been collected and automatically indexed over a twelve year period.

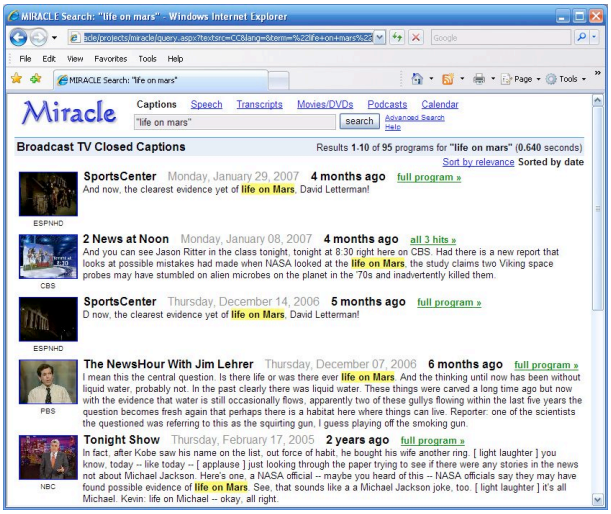

Figure 3. Results page with document excerpts for user query in *Miracle*

2.2 Concept computation

In this paper, *Miracle* was augmented to index low-level image similarity and semantic concepts. We use 374 classifier models (derived from the LSCOM lexicon [6]) trained over data in the TRECVID 2005 development set and published in [11]. TRECVID is an annual international evaluation of video processing tasks including shot boundary detection, high-level feature detection, and image search. TRECVID provides participants with a large collection of video data over which experiments are conducted and presented in an anual workshop. The training data is a set of over 64k TRECVID keyframes with positive and negative labels from multiple human annotators. Low-level features are extracted for three modalities: grid-based color moments, a global edge direction histogram, and global Gabor texture responses. Support vector machine (SVM) models were trained independently for each modality for each of the 374 concepts. In our work, we extract features for the three modalities above and apply the pre-trained models to a new, large set of images. It should be noted that we first resize all images to match the same size as the training data and use the same executables to extract and model low-level features. To derive a final score for each model, we average the output of the three modality classifiers. Thus, the final output of concept computation is a vector of 374 high-level concept scores for each keyframe in a video.

2.3 Concept system architecture

Once concept scores are computed, we must index the data in a robust and scalable form, illustrated in figure 4. Following the modular design of the *Miracle* platform, we store the concept scores for each image and similarity to other keyframes in individual XML files for each program. At the same time the XML file is created, we also insert all of the concept scores for a new video into an online database; in our implementation we used a MySQL backend. For fast searching, a single row in our table has only three columns: concept id, image id, and concept score. This combination of online and offline data allows a fast search of concept scores across the image database, while minimizing the amount of scores that must be loaded into memory from our xml-based file system.

2.3.1 Depth bounding for scalability

One problem that must be addressed when dealing with very large sets of data is that of storage and indexing. To optimally manage the scores from the 374 concepts, we keep only the top L scores for each concept in the online database. Not only is this a reasonable way to control database size, but given the fact that most users do not traverse a large number of search results (i.e. given 1000 results, most users will only browse the top 500), it is also an amicable solution aligned to user expectations. While the exact performance numbers may differ, we find that an L on the order of 10000 is acceptable.

2.3.2 Aggregate statistics

Aggregate statistics are derived from the sub-sampled concept scores described in the previous section. Statistics for the minimum, maximum, and average scores for

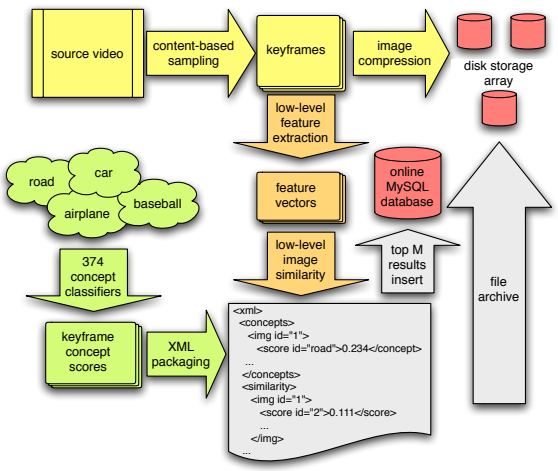

Figure 4. Visual illustration of *Miracle* concept processing framework.

each concept are computed and stored in individual rows in the online database. These statistics allow us to confidently construct score filters for each concept, instead of using ad-hoc thresholds and unstable heuristics. This approach also adheres to scalability requirements because as new captured data arrives (as is expected in the *Miracle* framework) the statistics can easily be recomputed during a non-critical time of day, but the overall concept filtering system will not suffer a significant performance loss if this update is delayed.

2.3.3 Suggested concepts

One critical component of our system is the ability to suggest concepts based on a given result set. We have visually illustrated a suggestions scenario in figures 5(a) and 5(d). Given any result set (perhaps from a text search, image similarity search, or even the results of a concept filter), we choose the top N images for analyis. For each concept, $i \in [1, C]$, we compute the fraction of results, π_i, that have a score greater than each concept's statistical average score, μ_i. To assist the user in understanding how the final suggested concept was chosen, we also output this count in the final user interface.

$$\pi_i = \frac{\sum_{n=1}^{N} count(s_{ni} \geq \mu_i)}{N}$$

Each result fraction is then scaled by the performance, α_i, of each respective concept by its performance on a labeled test set and then select the top M scoring concepts among all final scores $C_M \subset \{\alpha_1\pi_1, \ldots, \alpha_i\pi_i\}$.

We acknowledge that using the statistical mean of scores is prone to error for a classifier that has particularly poor performance. A more robust method for calculating a threshold for each concept could be derived by detecting the score where the biggest mutual information loss occurs. However, we defer this improvement for future iterations of the concept system.

2.3.4 Scoring filtered results

After the addition of a new concept filter, we must recompute the scores for the current set of results. Recomputing scores with a set of inclusive concept filters, $A \subset C$, is straightforward; simply average the scores s_{ni} of all inclusive concepts $i \in [1, A]$ for all result images $n \in [1, N]$.

$$s_n = \frac{\sum_{i=1}^{A} s_{ni}}{A}$$

Recomputing scores with a set of exclusive filters, $B \subset C$, is also simple, but instead of adding scores for excluded concepts $j \in [1, B]$, we conditionally remove result images n with excluded concept scores s_{nj} greater than the statistical mean score for that concept μ_j.

$$s_n = \begin{cases} \frac{\sum_{i=1}^{A} s_{ni}}{A} & (s_{nj} < \mu_j) \forall j \\ 0 & \text{otherwise} \end{cases}$$

For an in-depth discussion of filters an their impact on actual experiments, please see section 3.3.

2.4 User interface

For any truly useful interactive system, there must exist an intuitive and responsive user interface. For this purpose, we implemented a web-based interface using server-side scripting (PHP), cascading style sheets (CSS), and asynchronous web client requests (AJAX). Given the size of our dataset and the need to have high performance regardless of scale, we also integrated both file-based XML storage and online SQL indexing systems (see 2.3).

There are three major goals for our interface design: quickly expose the user to a variety of images, allow fast interaction with related images, and leverage concept scores to quickly guide the user towards concept filters that best partition the current set of images. Each of these goals is exemplified in parts of figure 5 and explained below.

Grid layout: A grid-based image layout allows users (experienced and novice) to quickly scan a high volume of content. While experiments in other works have demonstrated that high-speed image display is also powerful [4], we instead encourage deep interaction with result images. Additionally, for experienced users, we provide the ability to include concept-space and low-level similarity scores throughout the user interface. We assert that experts can use the relative differences in scores as they navigate a page

(a) interface overview (b) grid browsing results (c) related low-level images (d) concept filtering options

Figure 5. Example navigation using grid browsing, related images, and concept-based filtering.

of results to find discontinuities in scores that are otherwise lost in a grid-based display.

Related images: We recognize that a very powerful part of any image-query system is the ability to find visually related items. Towards this purpose, we allow the user to quickly inspect other images, ranked by similarity, without interrupting their browsing experience via asynchronous web requests to our pre-computed similarity XML files. In our current implementation, we only display similar images that reside in the same program; in future work, we plan to expand this display to include programs captured near the same date.

Suggested concept filters: Perhaps the most innovative feature of our user interface is the ability for users to quickly see how a set of images is related in the concept space, as described in section 2.3.3. Our interface provides a textual description of concepts that are identified as highly active given a set of user results. While the number of top ranking results to consider (N) and the number of active concepts to suggest (M) is adjustable, we defer discussion of ideal setting selection for future work and instead choose heuristic settings of N=20 and M=5 for our experiments. We chose N=20 so that only the "best" results returned in a search were considered for active concepts. We chose M=5 so that the user is not overwhelmed with choices for concept filtering and because after a certain depth of suggestions, the strength of the semantic relationship to the result images is too low.

Combining these three design components, we created the interface shown in figure 5. As our interface is merely a tool to explore a set of data, the search input can be a specific video, a set of results from another query (say a text query, as shown in figure 3), or simply the direct results from a concept search. Finally, we wish to guarantee that the current search state can be easily revisited in future uses of the system – either by the same user or another user assisting remotely in the search – so throughout the system we use absolute URLs that explicitly indicate the current search state. With this feature, users can easily revisit a set of concept filters that they have constructed so that after very similar video content is added to the index it is easily discovered. Finally, building on top of the *Miracle* system, we provide instant, in-depth content inspection in a list form, shown in figure 3, a grid-based timeline form, or with the content from a single keyframe streamed to the user at an appropriate bandwidth.

3 Experiment

We analyzed the performance of our concept filtering system over a subset of television broadcasts captured by *Miracle*. Our system allows users to quickly explore a large dataset within the semantic concept space. Even though the performance of some classifiers may not be perfect, our system enables user navigation through relevant semantic concepts that are dynamically exposed by the data, not traditional lexical approaches. The following sections describe the dataset, baseline classifier performance, example search scenarios, and measurements that demonstrate our system is very suitable for real-world scenarios.

3.1 Data subset

In this experiment, we chose one broadcaster (NBC) and analyzed content from video captured between January and June of 2006. During this time period five different programs were captured: *Meet the Press* (22) recordings, *NBC Nightly News* (158) recordings, *Channel 4 News at 6PM* (164) recordings, *The Apprentice* (13) recordings, and *The Tonight Show with Jay Leno* (91) recordings for a total of 357 discrete recordings and 314 hours of video. After processing these programs with *Miracle*'s content based sampling algorithm (described in 2.1), 385,873 keyframes were extracted with an average of 875 keyframes for each program. A quick survey of the generated keyframes indicates that there are some visual settings that are highly recurrent, such as the talk-show set of *The Tonight Show*, and some visual frames that are exact matches, commercials repeated in this segment. At this time, we do not cluster, group, or otherwise discriminate highly similar keyframes generated

concept filter	accuracy @ 40	accuracy @ 100
basketball only	0.475	0.360
+*athlete*	0.575	0.490
+*standing*	0.675	0.530
+*crowd*	0.825	*0.440*
baseball only	0.050	0.020
+*athlete*	0.175	0.25
+*sports*	0.375	0.340
-*soccer*	0.525	0.440
+*running*	0.600	0.540
+*grandstand*	*0.600*	*0.420*

Table 1. Accuracy of concept filtering iterations at depths of 40 and 100 results using only data-suggested concepts.

by the content based sampling algorithm. It should be noted that while the data in this experiment was derived from only one television channel, the diversity of the included programs demonstrates the system's applicability to any set of video data.

3.2 Classifier performance

Concept classifier performance varies for a number of reasons like labeler agreement, specificity of concept definition, number of training samples, and test data disparity, to name a few. While the first three variations are part of the training process discussed in [11], we would like to analyze the impact of data disparity on classifier performance. We conducted searches with two concepts from our lexicon: *basketball* and *baseball*. These two concepts were chosen because they contain a somewhat specific semantic (as opposed to *sky*) that does not exist as a single object (like *face* or *car*) and because the baseline concept detectors have both good (*basketball*) and poor (*baseball*) performance.

First, we explore the performance gains where a concept classifier already has quite high performance – the basketball concept. In table 1, we indicate accuracy at result depths of 40 images (the first browser page) and 100 images. The baseline detector has a relatively high accuracy at both sample points, indicating that there is a good fit between model and data. However, there are still several negative cases (indicated by red frames) in 6(a) caused by classifier error on one of our low-level modalities: color, texture, or edge histogram. Fortunately, the system immediately identifies a suitable set of suggested concepts that are strongly related to our current results: *athlete*, *standing*, and *crowd* (described in table 2). As we apply these filters individually, we can see accuracy increase while the visual relevance of suggested concepts decreases. This is particularly obvious when the third positive concept filter (*crowd*) is added and the accuracy at depth 100 drops. Fortunately, this result is expected as the suggested concepts like *swimmer* and *microphone* are quite unrelated to *basketball* and of the first 40 images, only 7 were unrelated.

concept filter	suggested concepts (# in top results)
basketball only	athlete (15), standing (9), scene_text (8), crowd (4), steeple (11)
+*athlete*	standing (8), scene_text (7), steeple (12), crowd (3), mountain (4)
+*standing*	scene_text (15), crowd (5), steeple (17), swimmer (3), dark-skinned_people (19)
+*crowd*	scene_text (10), steeple (14), dark-skinned_people (19), swimmer (2), microphones (2)
baseball only	cordless (4), canoe (2), athlete (2), underwater (1), entertainment (1)
+*athlete*	soccer (6), sports (9), grandstands (6), vegetation (7), lawn (7)
+*sports*	soccer (11), grandstands (14), running (9), lawn (11), vegetation (8)
-*soccer*	grandstands (12), running (9), scene_text (8), maps (4), swimming_pools (2)
+*running*	grandstands (15), scene_text (7), canoe (2), vegetation (5), natural-disaster (4)
+*grandstand*	scene_text (8), tennis (3), maps (3), vegetation (4), indoor_sports_venue (17)

Table 2. Analysis of data-suggested concepts during iterative concept filtering; concepts already selected are not listed again in suggested concepts.

Now, let us explore the performance gains of a concept with weaker initial performance, *baseball*. We observe in 6(d) and table 1 that the baseline detector finds only 2 positive samples at a depth of 40 images. We can infer that the green background of commercials and golf games has greatly polluted the baseline results. Fortunately, with the reasonable set of suggested concepts (figure 6) allow us start exploring the concept space of our result images. The story of performance gains are similar to that of the *basketball* concept, with measured accuracy at depths 40 and 100 increasing to some point and then tapering off, as shown by the italicized numbers in table 1. However, with the last concept filter applied, the accuracy jumps from the original 0.02 to the best 0.54 for baseball, which is both more significant than the *basketball* gain of 0.36 to 0.53 example, and a visually satisfying result to the user.

In both examples, it is clear that with several filters al-

(a) original 'basketball' concept (b) add 'athlete' concept (c) add 'standing' and 'crowd' concepts

(d) original 'baseball' concept (e) add 'athlete' and 'sports' concepts (f) exclude 'soccer' and add 'running' concepts

Figure 6. Example results for "basketball" and "baseball" concepts with iterative concept filtering.

ready applied, the impact subsequent filters is greatly reduced, but this is not surprising either, given our method for scoring results with multiple filters (section 2.3.4).

3.3 Utility of filtering

Traditionally, the performance gains from automatically filtering results or learning statistical concept relationships with concept scores are quite limited. Complications involving cut-off thresholds, independent concept performance, and the meaning of a low score are usually the main points of contention. However, through some empirical experiments (two described in this work), we demonstrate that in an interactive environment, concept filtering can dramatically improve search performance.

First, we address concerns for score thresholds and concept performance by collecting statistics over our given dataset. One may argue that this approach ignores the underlying question of performance because some obscure or highly variant concepts are just hard to generalize, but we assert that in a user-based, interactive setting the problem of low-quality concepts can be overcome *provided that the concept model was earnestly trained on a non-random set of images*. For example, in figure 6(d) and data in table 1, we observe that there are no instances of any baseball-related images in the top 20 results of the baseball concept. However, through iterative filtering and the combination of other concepts, baseball-related images are eventually moved to high ranking positions. Second, concerns about low concept scores can be answered in part by our approach, which is ambivalent to the actual score range for a concept and instead relies on collected statistics. While a numerically high scores *does* imply relevance for a concept, the relationship is non-symmetric and a numerically low score for a concept is better likened to a noise or background model.

3.4 Execution profile

Although some processing steps in this concept system can be time-prohibitive, the majority of this time is consumed by pre-processing and not searching or indexing; this reflects our goal of performance optimization in repeated user tasks, not off-line machine computation. First, the labeling of images and training of the concept models is the most time consuming task. Quoting the original technical report for these models, which was trained on 2GHz single-thread machines, "running even such a light-weight training process for all 374 concepts takes approximately 3 weeks using 20 machines in parallel, or roughly more than a year of machine time," [11]. This time was estimated with non-optimized, java-based execution and used an exhaustive SVM parameter search for each modality model. Second, the capture of new broadcast content and content-based sampling (to generate image keyframes) is real-time and is constrained only by the capture process. Third, execution time for low-level feature extraction and concept scoring for all 374 models occurs in roughly 0.16x and 0.78x real-time respectively, when executed on 3.2GHz multi-core ma-

chines. Adding computation results to the online database and file repository is on the order of seconds and an infrequently needed update of aggregate statistics (described in 2.3.2) completes all updates in under 5 minutes, which is also dependent on the depth bound. Finally, a search of the database (see 2.3.1) during an interactive session completes in 5-10 seconds depending on the number of concept filters applied, where subsequent searches for browsing deep into the result list complete in about 1 second.

4 Conclusion

Semantic concept search is a very difficult task that requires a robust set of features, concept models, and a well-tempered approach for searching. Unavailable to automatic search methods, we leverage highly responsive user interaction to quickly traverse search results in the semantic concept space. Building on other work that demonstrates the benefit of concepts over low-level features alone, we provide a system that works well even with relatively poor performance of individual concept classifiers.

The gradual improvements using semantic filters seen in figure 6 demonstrate three important points. First, the use of semantic concepts extends image search beyond what is capable using content-based image retrieval because they tolerate a diversity of scenes and objects. Second, while exploring the semantic space of a set of results, users can discover unique, data-driven relationships that were otherwise unexpected when compared to traditional semantic approaches that rely on lexical similarity or a hand-crafted set of rules and relationships. The user no longer needs to be familiar with the lexicon of a concept space or how each concept is ontologically related. Finally, we allow users to join several concept filters that can either include or exclude results from the returned set, which can be saved via a unique search URL and explored again when new data is indexed with the *Miracle* processing system.

4.1 Future work

This work is one of the first to enable interactive concept space exploration over a very large set of diverse images. First, to position our classifier models at the state of the art, much more complex low-level features and machine learning frameworks can be applied [2]. However, we believe the most promising improvements will include powerful, potentially domain specific features like face detectors and motion estimation features that account for temporal aspects of the data. Second, we plan to extend frame-based similarity computations to include other captured broadcasts around the same time span. This way, we can more accurately identify duplicates like commercials and or near duplicate material inter-program segments (like weather announcements in news broadcasts). Third, additional experiments should be conducted to find the ideal L, M, and N values that were heuristically determined in our current implementation. Unfortunately, these experiments would require a large amount of user interaction data, which requires significant time and effort to collect. Similarly, we hope to find a more data-sensitive method for computing a threshold value for each concept (currently derived from the statistical mean of a concept's scores. Finally, we also plan to incorporate some form of active learning or relevance feedback that would allow the adaptation of existing models to new data captured by *Miracle*.

References

[1] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2, 2001.

[2] J. Cao, Y. Lan, J. Li, Q. Li, X. Li, F. Lin, X. Liu, L. Luo, W. Peng, D. Wang, et al. Intelligent Multimedia Group of Tsinghua University. In *NIST TRECVID workshop*, Gaithersburg, MD, 2006.

[3] D. Gibbon, Z. Liu, and B. Shahraray. The MIRACLE system. *CCNC 2006*, Jan. 8-10, 2006.

[4] A. Hauptmann, W. Lin, R. Yan, J. Yang, and M. Chen. Extreme video retrieval: joint maximization of human and computer performance. *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 385–394, 2006.

[5] W. Jiang, S.-F. Chang, and A. C. Loui. Context-based concept fusion with boosted conditional random fields. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hawaii, USA, April 2007.

[6] L. Kennedy. Revision of LSCOM Event/Activity Annotations, DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia. Technical report, Columbia University, December 2006.

[7] Z. Liu, D. Gibbon, and B. Shahraray. Multimedia Content Acquisition and Procesing in the MIRACLE system. *CCNC 2006*, Jan. 8-10, 2006.

[8] Z. Liu, D. Gibbon, E. Zavesky, B. Shahraray, and P. Haffner. A Fast, Comprehensive Shot Boundary Determination System. *Multimedia and Expo, 2007. ICME'07. Proceedings. 2007 International Conference on*, 2007.

[9] B. Shahraray. Scene Change Detection and Content-based Sampling of Video Sequence. *SPIE 2419*, February 1995.

[10] J. R. Smith and S.-F. Chang. Visualseek: a fully automated content-based image query system. In *ACM Multimedia*, Boston, MA, Nov 1996.

[11] A. Yanagawa, W. Hsu, and S.-F. Chang. Brief descriptions of visual features for baseline trecvid concept detectors. Technical report, Columbia University, July 2006.

[12] Q. Zhang, S. Goldman, W. Yu, and J. Fritts. Content-Based Image Retrieval Using Multiple-Instance Learning. *Proceedings of the Nineteenth International Conference on Machine Learning table of contents*, pages 682–689, 2002.