IBM Research TREC-2002 Video Retrieval System

Bill Adams[*], Arnon Amir[†], Chitra Dorai[‡], Sugata Ghosal[§], Giridharan Iyengar[*], Alejandro Jaimes[‡], Christian Lang[‡], Ching-yung Lin[‡], Apostol Natsev[‡], Milind Naphade[‡], Chalapathy Neti[*], Harriet J. Nock[*], Haim H. Permuter[†], Raghavendra Singh[§], John R. Smith[‡], Savitha Srinivasan[†], Belle L. Tseng[‡], Ashwin T. V.[§], Dongqing Zhang[¶]

Abstract

In this paper, we describe the IBM Research system for analysis, indexing, and retrieval of video, which was applied to the TREC-2002 video retrieval benchmark. The system explores novel methods for fully-automatic content analysis, shot boundary detection, multi-modal feature extraction, statistical modeling for semantic concept detection, and speech recognition and indexing. The system supports querying based on automatically extracted features, models, and speech information. Additional interactive methods for querying include multiple-example and relevance feedback searching, cluster, concept, and storyboard browsing, and iterative fusion based on user-selected aggregation and combination functions. The system was applied to all four of the tasks of the video retrieval benchmark including shot boundary detection, concept detection, concept exchange, and search. We describe the approaches for each of the tasks and discuss some of the results.

1 Introduction

The growing amount of digital video is driving the need for more effective methods for indexing, searching, and retrieving video based on its content. Recent advances in content analysis, feature extraction, and classification are improving capabilities for effectively searching and filtering digital video content. Furthermore, the recent MPEG-7 standard promises to enable interoperable content-based retrieval by providing a rich set of standardized tools for describing features of multimedia content [1]. However, the extraction and use of MPEG-7 descriptions and the creation of usable fully-automatic video indexing and retrieval systems remains a significant technical challenge.

The TREC video retrieval benchmark is facilitating the technical advancement of content-based retrieval of video by standardizing a benchmark video corpus along with different video retrieval and detection tasks. The benchmark provides a consistent evaluation framework for assessing progress as researchers experiment with novel video indexing techniques. This year, we participated in the TREC video retrieval benchmark and submitted results for four tasks: (1) shot boundary detection, (2) concept detection, (3) concept exchange, (4) search. We explored several diverse methods for video analysis, indexing, and retrieval, which included automatic descriptor extraction, statistical modeling, and multi-modal fusion. We conducted experiments that individually explored audio-visual and speech modalities as well as their combination in manual and interactive querying. In the paper, we describe the video indexing and retrieval system and discuss the results on the video retrieval benchmark.

1.1 Outline

The outline is as follows: in Section 2, we describe our process for video and speech indexing. In Section 3, we describe the video retrieval system including methods for content-based search, model-based search, speech-based search, and other methods for interactive searching and browsing. In Section 4, we discuss the approaches for each of the benchmark tasks and examine some of the results.

2 Video indexing system

The video indexing system analyzes the video in an off-line process that involves video content indexing and speech indexing. The video content indexing process consists of shot boundary detection, key-frame extraction, feature extraction, region extraction, concept detection, and clustering, as shown in Figure 1. The basic unit of indexing and retrieval is a video shot.

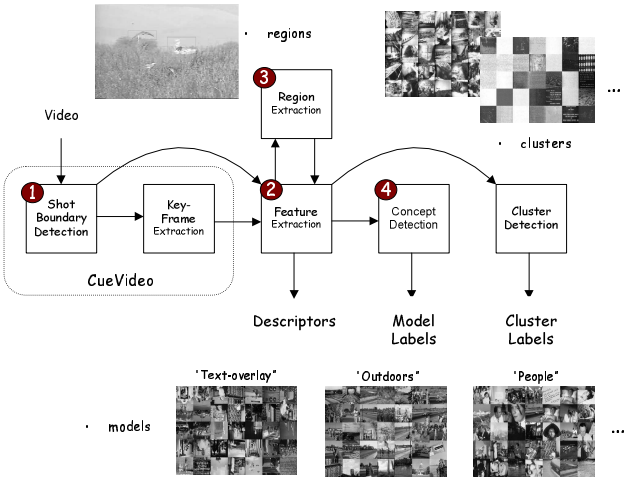

Figure 1: Summary of video content indexing process.

[*]IBM T. J. Watson Research Center, 1101 Kitchawan Rd., Yorktown Heights, NY 10598

[†]IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120

[‡]IBM T. J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532

[§]IBM India Research Lab, Block 1, IIT, New Delhi 110016, India

[¶]Dept. of E.E., Columbia University, New York, NY 10027

2.1 Shot boundary detection (SBD)

Shot boundary detection (SBD) is performed using the real-time *IBM CueVideo* system [2] which automatically detects shots and extracts key-frames. This year, we explored several methods for making SBD more robust to poor video quality. Some of the methods include using localized edge gradient histograms and comparing pairs of frames at greater temporal distances. Overall, our 2002 SBD system showed reduction in errors by more than 30% compared to our 2001 SBD system [3].

The baseline CueVideo SBD system uses sampled, three-dimensional color histograms in RGB color space to compare pairs of frames. Histograms of recent frames are stored in a buffer to allow a comparison between multiple image pairs up to seven frames apart. Statistics of frame differences are computed in a moving window around the processed frame and are used to compute the adaptive thresholds, shown in Figure 2 as a line above the difference measures (Diff1, Diff3 and Edge1). A state machine is used to detect the different events (states). The SBD system does not require any sensitivity-tuning parameters. More details about the baseline system can be found in [3, 4].

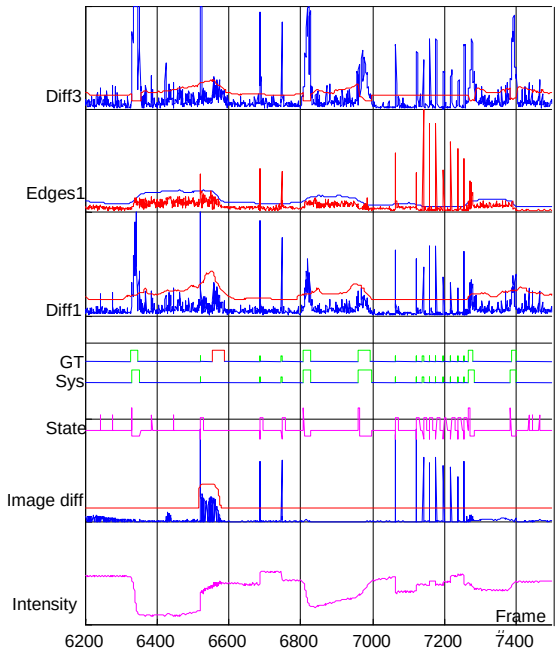

Figure 2: Plot of frame-to-frame processing of the SBD algorithm. Notice the ground truth (GT) and system output (Sys) plots for this segment of video which has six dissolves (one missed) and twelve cuts.

Several changes were incorporated to the baseline SBD algorithm to accommodate lower video quality, as was the case for the videos in the TREC-02 data set. Localized edge-gradient histograms were added to overcome color errors. The 512-bin edge-gradient histogram counts the number of pixels in each of eight image regions, having similar I_x, I_y derivatives (each derivative is quantized into three bits). Thus it is less sensitive to lighting and color changes. Rank filtering was added in time/space/histogram at various different points along the processing to handle the new types and higher levels of noise. The comparison of pairs of frames at wider distances up to thirteen frames apart was added to overcome the high MPEG-1 compression noise. Several new states were added to the state machine to detect certain types of video errors and to detect very short dissolves that were 2-3 frames long. These changes were tuned based on precision-recall measurements using data subsets from TREC01 test set and TREC02 training set.

2.2 Feature extraction

The system extracts a number of descriptors for each video shot. Some of the descriptors, as indicated below, are extracted in multiple ways from each key-frame image using different normalization strategies (see [5]) as follows: (1) global, (2) 4x4 grid, (3) 5-region layout, and (4) automatically extracted regions. The following descriptors were extracted:

- Color histogram (global per key-frame, 4x4 grid, 5-region layout, segmentation regions): one based on a 166-bin HSV color space [6] and another based on 512-bin RGB color space,
- Color correlogram (global per key-frame, 4x4 grid, 5-region layout): based on a single-banded auto-correlogram coefficients extracted for 8 radii depths in 166-color HSV color space [7],
- Edge orientation histogram (global per key-frame, 4x4 grid, 5-region layout): based on Sobel filtered image and quantization to 8 angles and 8 magnitudes [5],
- Wavelet texture (global per key-frame, 4x4 grid, 5-region layout): based on wavelet spatial-frequency energy of 12 bands using quadrature mirror filters [6],
- Tamura texture (global per key-frame, segmentation regions): Three values representing the coarseness, contrast, and directionality, respectively [8],
- Co-occurrence texture (global per key-frame, 4x4 grid, 5-region layout): based on entropy, energy, contrast, and homogeneity features extracted from gray-level co-occurrence matrices at 24 orientations [9],
- Motion vector histogram (global per shot, segmentation regions): based on 8×8 motion estimation blocks in the MPEG-1 decoded I and P frames. A six-bin histogram is generated based on the motion vector magnitudes,
- Mel-Frequency Cepstral Coefficients (MFCC): transformation of uncompressed PCM signal to 24 MFCC features including the energy coefficient.

2.3 Region extraction

In order to better extract local features and detect concepts, we developed a video region segmentation system that automatically extracts foreground and background regions from video. The system runs in real-time with extraction of regions from I-frames and P-frames in MPEG-1 video. The segmentation of the background scene regions uses a block-based region growing method based on color histograms, edge histograms, and directionality. The segmentation of the foreground regions uses a spiral searching technique to calculate the motion vectors of I- and P- frames. The motion features are used in region growing in the spatial domain with additional tracking constraints in the time domain. Although

we tested MPEG-1 compressed-domain motion vectors, we found them to be too noisy. We also found that combining motion vectors, color, edge, and texture information for extraction of foreground objects did not give significantly better results than using only motion.

2.4 Clustering

We used the extracted visual descriptors (see Section 2.2) to cluster the video shots into perceptually similar groups. We used a k-means clustering algorithm to generate 20 clusters. We found color correlograms to achieve an excellent balance between color and texture features. The clusters were later used to facilitate browsing and navigation for interactive retrieval (as described in Section 3.5).

2.5 Concept detection

The concept detection system learns from labeled training video content to classify unknown video content (in our case, the feature test and search test data). We have investigated several different types of statistical models including Support Vector Machines (SVM), Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM).

2.5.1 Lexicon design

The first step in designing a semantic concept detection system is the construction of a concept lexicon [10]. We viewed the training set video and identified the most salient frequently occurring concepts and fixed a lexicon of 106 concepts, which included the 10 concepts belonging to the TREC concept detection task (denoted as primary concepts). Overall, we generated training and validation data and modeled the following 10 primary concepts: Outdoors, Indoors, Cityscape, Landscape, Face, People, Text Overlay, Music, Speech and Monologue. We also modeled the following 39 secondary generic concepts:

- Objects: Person, Road, Building, Bridge, Car, Train, Transportation, Cow, Pig, Dog, Penguin, Fish, Horse, Animal, Tree, Flower, Flag, Cloud,
- Scenes: Man Made Scenes, Beach, Mountain, Greenery, Sky, Water, Household Setting, Factory Setting, Office Setting, Land, Farm, Farm House, Farm Field, Snow, Desert, Forest, Canyon,
- Events: Parade, Explosion, Picnic, Wedding.

2.5.2 Annotation

In order to generate training and validation data, we manually annotated the video content using two annotation tools[1] – one produced the visual annotations and the other produced audio annotations. The IBM MPEG-7 Video Annotation Tool (*a.k.a. VideoAnnEx*), shown in Figure 3, allows the shots in the video to be annotated using terms from an imported lexicon. The tool is compatible with MPEG-7 in that the lexicons can be imported as MPEG-7 classification schemes and generates MPEG-7 descriptions of the video based on the detected shots and annotations. The tool also allows the users to directly create and edit lexicons.

Figure 3: VideoAnnEx MPEG-7 video annotation tool. The system enables semi-automatic annotation of video shots and editing of the lexicon.

The second tool, the IBM Multimodal Annotation Tool, provides three modes of annotation: video, audio with video, or audio without video. The audio annotation is based upon audio segments in which the user manually delimits each segment within the audio upon listening and selects from the lexicon those terms that describe the audio content. Multimodal concepts (e.g. Monologues) are annotated using audio with video mode of annotation.

2.5.3 Concept modeling

Semantic concept detection was investigated using a statistical classification methodology (as described in [11, 12, 10]). The system learns the parameters of the classifiers using training data for each concept using statistical methods. We considered two approaches: one based on a decision theoretic approach and the other based on a risk minimization approach.

Decision theoretic approach In this approach, the descriptors are assumed to be independent identically distributed random variables drawn from known probability distributions with unknown deterministic parameters. For the purpose of classification, we assume that the unknown parameters are distinct under different hypotheses and can be estimated.

Structural risk minimization Unlike the decision theoretic approach, the discriminant approach focuses only on those characteristics of the feature set that discriminate between the two hypotheses of interest. The idea of constructing learning algorithms based on the structural risk minimization inductive principle was proposed in [13]. In particular, we used Support Vector Machines (SVM)[2], which map the feature vectors into a higher dimensional space through nonlinear function and constructing the optimal separating hyper-plane.

Training and validation Training and validation of models was done using the NIST feature training data set. We randomly partitioned the NIST feature training data set into a 19 hour Feature

[1] Annotation tools are available at http://alphaworks.ibm.com

[2] We used SVMLight toolkit (http://svmlight.joachims.org/)

Training (FTR) collection and a 5 hour Feature Validation (FV) collection. We used the FTR collection to construct the models and the FV collection to select parameters and evaluate the concept detection performance. The validation process was beneficial in helping to avoid over-fitting to the FTR collection.

2.5.4 Fusion

Since no single descriptor is powerful enough to encompass all aspects of video content and separate the concept hypotheses, combining information is needed at several levels in the concept modeling and detection processes. We experimented with two distinct approaches involving early fusion and late fusion. For early fusion we experimented with fusing descriptors prior to classification. For late fusion we experimented with retaining soft decisions and fusing classifiers. In addition, we explored various combining methods and aggregation functions for late fusion of search results as described in Section 3.6. Two modeling procedures are used. They use different subsets of visual features. The first procedure utilizes both early and late fusions, while the second procedure uses only late fusion.

Feature fusion The objective of feature fusion is to combine multiple features at an early stage to construct a single model. However, since this increases the dimensionality of the feature space—which makes it sparser—it also makes the classification problem harder and increases the risk of over-fitting the data. This approach is therefore most suitable for concepts that have sufficiently large number of training set examples that would allow the classifier to exploit correlations between the features. We experimented with feature fusion by simply normalizing and concatenating descriptors. Different combinations of descriptors were used to construct models. We used the validation set to choose the best combination.

Classifier fusion In an ideal situation, early fusion should work for all concepts, since all of the information is available to the classifier. However, practical considerations, such as limited number of training examples and the increased risk of over-fitting necessitate an alternate strategy. If the features are fairly de-correlated, then treating them independently is less of a concern. In such situations, we model concepts in each modality or feature space independently, and fuse individual classifier decisions later. We used a separate model (SVM or GMM) for each descriptor, which results in multiple classifications and associated confidences for each shot depending on the descriptor. While the classifiers can be combined in many ways, we explored normalized ensemble fusion to improve overall classification performance.

2.5.5 Specialized detectors

Although we used the above generic approaches for detection of most concepts, for two concepts (monologues and text overlay) we explored specialized approaches as follows:

Monologue detection For monologue detection, we first performed speech and face detection on each shot. Then, for shots containing speech and face, we further evaluated the synchrony between the face and speech using mutual information and used the combined score thus generated to rank all shots in the corpus. Based on experimental results of a variety of synchrony detection techniques, we used a scheme that models audio and video features as locally Gaussian distributions (see [14] for more details).

Text overlay detection We explored two algorithms for extracted overlay text in video and fused the results of the classifiers to produce the final concept labeling. The first method (see [15]) works by extracting and analyzing regions in a video frame. The processing stages in this system are: (1) isolating regions that may contain text characters, (2) separating each character region from its surroundings and (3) verifying the presence of text by consistency analysis across multiple text blocks. A confidence measure is computed as a function of the number of characters in text objects in the frame. The second method uses macro-block-based texture and motion energy. Layout analysis is used to verify the layout of these character blocks. A text region is identified if the character blocks can be aligned to form sentences or words.

2.6 Speech recognition and indexing

As in TREC-2001, we constructed a speech-based system for video retrieval. Significant improvements were made to both the automatic speech recognition (ASR) performance and the speech search engine performance relative to our TREC-2001 submission.

2.6.1 Automatic speech recognition (ASR)

A series of increasingly accurate speech transcriptions for the entire corpus were produced in the period leading up to the evaluation. The first set of transcriptions were produced using an IBM real-time transcription system tuned for Broadcast News; this is the same transcription system as was used in TREC-2001 [3]. Later transcriptions were produced using an off-line, multiple pass transcription system comprising the following stages (see [16] for more details and citations):

- Remove silent videos
- Divide each video into segments using Bayesian Information Criteria (BIC)
- Detect "music" and "silence" and transcribe using an IBM $10\times$Real-Time Broadcast News transcription System
- Apply supervised Maximum Likelihood Linear Regression (MLLR) adaptation of speaker-independent HUB4 models using a set of eight (word-level transcribed) videos
- Decode "speech-only" segments using interpolated trigram Language Model (LM)
- Cluster "speech-only" segments into "speaker- and environment- similar" clusters
- Apply unsupervised MLLR adaptation of TREC-2002-adapted HUB4 models to each cluster using single global MLLR mean and precision transforms

The word error rate (WER) of the final transcripts is estimated at 34.6% on a held out set of six videos from Search Test and Feature Test which were manually transcribed[3]. This compares favorably to 39.0% for the best of the publicly-released transcriptions on the same set and represents a 41% improvement over the transcriptions used as the basis for IBM's TREC-2001 SDR system.

[3]Note this set does not overlap with the set used in supervised acoustic model adaptation.

2.6.2 Speech indexing

Indexes were constructed for SDR from the final most accurate speech transcriptions. Three types of indexes were generated: document-level indexes, an inverse word index, and a phonetic index. No attempt was made to index the set of silent videos.

Document-level indexes: The document-level indexes support retrieval at the document level, where a document is defined to span a temporal segment containing at most 100 words[4]. Consecutive documents overlap by 50 words in order to address boundary truncation effects. Once documents are defined and their associated time boundaries are recorded, the documents are preprocessed using (1) tokenization to detect sentence/phrase boundaries; (2) (noisy) part-of-speech tagging such as noun phrase, plural noun etc; (2) morphological analysis, which uses the part-of-speech tag and a morph dictionary to reduce each word to its morph eg. verbs `[lands]`, `[landing]` and `[land]` reduce to `/land/`; (4) "stop" words are removed using standard stop-word lists. After pre-processing, indexes are constructed and statistics (such as word and word pair term- and inverse-document frequencies) are recorded for use during retrieval.

Inverse word index: the inverse word index supports Boolean search by providing the $(video_i, time_i)$ of all the occurrences of a query term in the videos. Preprocessing of transcripts is similar to that above.

Phonetic index: the phonetic index supports search of out-of-vocabulary words. The (imperfect) speech transcript is converted to a string of phones [17]. The phonetic index can be searched for sound-like phone sequences, corresponding to out-of-vocabulary query terms such as some acronyms, names of people, places, and so forth[5]

3 Video retrieval system

The video retrieval system provides a number of facilities for searching, which include content-based retrieval (CBR), model-based retrieval (MBR), speech-based search or spoken document retrieval (SDR) and other interactive methods.

3.1 Content-based retrieval (CBR)

The objective of CBR is to match example query content to target video content using the extracted descriptors (see Section 2.2). The degree of match is determined on basis of feature similarity, which we have measured using Minkowski-form metrics considering values of $r = 1$ (Manhattan distance) and $r = 2$ (Euclidean distance) as follows: given descriptors represented as multi-dimensional feature vectors, $\mathbf{v}_q$ and $\mathbf{v}_t$ be the query and target vectors, respectively, then

$$d^r_{q,t} = (\sum_{m=0}^{M-1} |v_q[m] - v_t[m]|^r). \quad (1)$$

3.2 Model-based retrieval (MBR)

Model-based search allows the user to retrieve video shots based on the concept labels produced by the models (see Section 2.5). In MBR, the user enters the query by typing label text, or the user selects from the label lexicon. Since a confidence score is associated with each automatically assigned label, MBR ranks the shots using a distance $\mathcal{D}$ derived from confidence $\mathcal{C}$ using $\mathcal{D} = 1 - \mathcal{C}$.

3.3 Speech-based search (SDR)

Speech-based search allows the user to retrieve video shots based on the speech transcript associated with the shots. We used multiple SDR systems independently and combined the results to produce the final SDR results for TREC-2002; we refer to the three systems as OKAPI-SYSTEM-1, OKAPI-SYSTEM-2, BOOLEAN-SYSTEM-1. To evaluate different design decisions, a limited ground truth was created for the combined FTR and FV collections by pooling the results and performing relevance assessment.

Query development and preprocessing: All SDR systems operate using a textual statement of information need. Query strings are pre-processed in a similar manner to the documents: tokenization, tagging and morphing gives the final query term sequence for use in retrieval.

Video segment retrieval: Given a query, the three SDR systems rank documents or video segments as follows:

- OKAPI-SYSTEM-1, OKAPI-SYSTEM-2: a single pass approach is used to compute a relevancy score for each document. Each document is ranked against a query, where the relevancy score is given by the OKAPI formula [18]. The total relevancy score for the query string is the combined score of each of the query terms. The scoring function takes into account the number of times each query term occurs in the document and how rare that query term is across the entire corpus, with normalization based upon the length of the document to remove the bias towards longer documents since longer documents are more likely to have more instances of any given word.
- BOOLEAN-SYSTEM-1: a Boolean search was applied to Boolean queries. This search also supported phonetic search of out-of-vocabulary words using the phonetic index, in conjunction with in-vocabulary words which can be located in the inverse word index.

Many SDR systems use the results of first pass retrieval as the basis for automatic query expansion scheme prior to running a second pass of retrieval. Experiments showed little gain from using an LCA-based scheme [19] on FTR+FV, since the number of relevant documents retrieved per query in the first pass is quite low, so the approach was not investigated further.

[4]Minor differences in document definition were used in constructing the different indexes, such as whether or not document boundaries are defined at long stretches of silence or music; experiments suggest these differences do not make a significant contribution to the differences in MAP across systems.

[5]For this year's queries we found the phonetic index was of limited use: only two queries involved out-of-vocabulary words, which were names.

Video segment-to-shot mapping: NIST evaluates video retrieval performance at the level of shots, rather than at the level of documents or video segments which span one or more shots. Thus we must somehow use the scores assigned to documents or video segments by SDR to assign scores at the level of shots[6]. The mappings used in the three component systems are:

- OKAPI-SYSTEM-1: the score assigned to a document is assigned to the longest shot overlapping that document;
- OKAPI-SYSTEM-2: the score assigned to a document is assigned to all the overlapping shots. A slightly higher score given to the later shots than to the first ones;
- BOOLEAN-SYSTEM-1: First, the boundaries of the video segment are determined by the coverage of the relevant words. Then the overlapping shots are scored the same way as with OKAPI-SYSTEM-2.

The video segment-to-shot mapping is critical to overall SDR performance. Post-evaluation experiments show the schemes above were not optimal choices; for example, since multiple relevant shots often overlap a single document, OKAPI-SYSTEM-1 performance can be improved simply by assigning a document score to all overlapping shots. Our current research is investigating more sophisticated schemes.

Fusion of multiple SDR systems: Analysis of the results from the different systems shows that they are often complementary on FTR+FV: no system consistently outperforms the others. Thus we hypothesized fusion of scores might lead to improved overall performance. Whilst various fusion schemes are possible, for TREC-2002 we use a simple additive weighted scheme to combine shot-level, zero-to-one range normalized scores from each of our basic SDR systems. Weights can be optimized on FTR+FV prior to the final run on (held-out) search test data. This combined system is termed "SDR-FUSION-SYSTEM".

3.4 Term vector search

We used term vectors constructed from the ASR text for allowing similarity search based on textual content. Given the entire collection of shots, we obtained a list of all of the distinct terms that appear in the ASR for the collection. The order of this list was fixed to give a one-to-one mapping of distinct terms and dimensions of the vector space. Each shot was then represented by an n-dimensional vector, where the value at each dimension represented the frequency of the corresponding term in each shot. This allows the comparison of two shots based on frequency of terms. We constructed several term vector representations based on ASR-text.

3.5 Browsing and navigation

The system provides several methods for browsing and navigation. For each video a story-board overview image was generated that allowed its content to be viewed at a glance. The system also generated these overview images for each cluster (see Section 2.4) and each model (see Section 2.5).

[6]Whilst this procedure might be simplified by defining documents in a fashion more closely related to shot boundaries, our results to date have found this to be less successful than the approaches discussed above.

3.6 Iterative fusion

The interactive fusion methods provide a way for combining and rescoring results lists through successive search operations using different combination methods and aggregation functions defined as follows:

Combination methods Consider results list R_k for query k and results list Q_r for current user-issued search, then the combination function $R_{i+1} = \mathcal{F}_c(R_i, Q_r)$ combines the results lists by performing set operations on list membership. We explored the following combination methods:

- Intersection: retains only those items present in both results lists.
$$R_{i+1} = R_i \cap Q_r \tag{2}$$
- Union: retains items present in either results list.
$$R_{i+1} = R_i \cup Q_r \tag{3}$$

Aggregation functions Consider scored results list R_k for query k, where $D_k(n)$ gives the score of item with id = n and $Q_d(n)$ the scored result for each item n in the current user-issued search, then the aggregation function re-scores the items using the function $D_{i+1}(n) = \mathcal{F}_a(D_i(n), Q_d(n))$. We explored the following aggregation functions:

- Average: takes the average of scores of prior results list and current user-search. Provides "and" semantics. This can be useful for searches such as "retrieve items that are indoors *and* contain faces."
$$D_{i+1}(n) = \frac{1}{2}(D_i(n) + Q_d(n)) \tag{4}$$
- Minimum: retains lowest score from prior results list and current user-issued search. Provides "or" semantics. This can be useful in searches such as "retrieve items that are outdoors *or* have music."
$$D_{i+1}(n) = \min(D_i(n), Q_d(n)) \tag{5}$$
- Maximum: retains highest score from prior results list and current user-issued search.
$$D_{i+1}(n) = \max(D_i(n), Q_d(n)) \tag{6}$$
- Sum: takes the sum of scores of prior results list and current user-search. Provides "and" semantics.
$$D_{i+1}(n) = D_i(n) + Q_d(n) \tag{7}$$
- Product: takes the product of scores of prior results list and current user-search. Provides "and" semantics and better favors those matches that have low scores compared to "average".
$$D_{i+1}(n) = D_i(n) \times Q_d(n) \tag{8}$$
- A: retains scores from prior results list. This can be useful in conjunction with "intersection" to prune a results list, as in searches such as "retrieve matches of beach scenes but retain only those showing faces."
$$D_{i+1}(n) = D_i(n) \tag{9}$$
- B: retains scores from current user-issued search. This can be useful in searches similar to those above but exchanges the arguments.
$$D_{i+1}(n) = Q_d(n) \tag{10}$$

3.7 Normalization

The normalization methods provide a user with controls to manipulate the scores of a results list. Given a score $D_k(n)$ for each item with id = n in results set k, the normalization methods produce the score $D_{i+1}(n) = \mathcal{F}_z(D_i(n))$ for each item n as follows:

- Invert: Re-ranks the results list from bottom to top. Provides "not" semantics. This can be useful for searches such as "retrieve matches that are *not* cityscapes."

$$D_{i+1}(n) = 1 - D_i(n) \tag{11}$$

- Studentize: Normalizes the scores around the mean and standard deviation. This can be useful before combining results lists.

$$D_{i+1}(n) = \frac{D_i(n) - \mu_i}{\sigma_i}, \tag{12}$$

  where μ_i gives the mean and σ_i the standard deviation, respectively, over the scores $D_i(n)$ for results list i.

- Range normalize: Normalizes the scores within the range $0 \ldots 1$.

$$D_{i+1}(n) = \frac{D_i(n) - \min(D_i(n))}{\max(D_i(n)) - \min(D_i(n))} \tag{13}$$

3.8 Shot expansion

The shot expansion methods allow the user to expand a results list to include for each shot its temporally adjacent neighbors. This can be useful in growing the matched shots to include a larger context surrounding the shots, as in searches such as "retrieve shots that surround those specific shots that depict beach scenes."

3.9 Multi-example search

Multi-example search allows the user to provide or select multiple examples from a results list and issue a query that is executed as a sequence of independent searches using each of the selected items. The user can also select a descriptor for matching and an aggregation function for combining and re-scoring the results from the multiple searches. Consider for each search k of K independent searches the scored result $S_k(n)$ for each item n, then the final scored result $Q_d(n)$ for each item with id = n is obtained using a choice of the following fusion functions:

- Average: Provides "and" semantics. This can be useful in searches such as "retrieve matches similar to item "A" *and* item "B".

$$Q_d(n) = \frac{1}{K} \sum_k (S_k(n)) \tag{14}$$

- Minimum: Provides "or" semantics. This can be useful in searches such as "retrieve items that are similar to item "A" or item "B".

$$Q_d(n) = \min_k (S_k(n)) \tag{15}$$

- Maximum:

$$Q_d(n) = \max_k (S_k(n)) \tag{16}$$

- Sum: Provides "and" semantics.

$$Q_d(n) = \sum_k (S_k(n)) \tag{17}$$

- Product: Provides "and" semantics and better favors those items that have low scoring matches compared to "average".

$$Q_d(n) = \prod_k (S_k(n)) \tag{18}$$

3.10 Relevance feedback search

Relevance feedback based search techniques enhance interactive search and browsing. The user's feedback on a set of shots is used to refine the search and retrieve in minimum number of iterations the desired matches. The user implicitly provides information about the matches being sought or *query concept* by marking whether shots are relevant or non-relevant in relation to his/her desired search output. The system utilizes this feedback to learn and refine an approximation to the user's *query concept* and retrieve more relevant video-clips in the next iteration.

We use a robust relevance feedback algorithm [20] that utilizes non-relevant video-clips to optimally delineate the relevant region from the non-relevant one, thereby ensuring that the relevant region does not contain any non-relevant video-clips. A similarity metric estimated using the relevant video-clips is then used to rank and retrieve database video-clips in the relevant region. The partitioning of the feature space is achieved by using a piecewise linear decision surface that separates the relevant and non-relevant video-clips. Each of the hyper-planes constituting the decision surface is normal to the minimum distance vector from a non-relevant point to the convex hull of the relevant points. With query concepts that can reasonably be captured using an ellipsoid in the feature space. The proposed algorithm gives a significant improvement in precision as compared to simple re-weighting and SVM-based relevance feedback algorithms.

4 Tasks and results

We participated four tasks: shot boundary detection (SBD), concept detection, concept exchange, and search.

4.1 Shot boundary detection (SBD) results

For the shot boundary detection task, the results of five systems were submitted, one of which was last year's SBD system as a baseline. A large difference in performance relative to last year was anticipated due to the degraded video quality of the TREC '02 data. The other four were different versions of the improved system, mainly applying different logic to the fusion of color histogram and the localized edges histogram information. Three of them performed well and yielded very similar results, while the forth one did not perform as well. Table 1 summarizes the evaluation of the baseline system, *alm1*, and the best new system, *sys47*, on last year's and this year's TREC video data test sets. The results on TREC-01 data set were computed by us, while the results for the TREC-02 data set are taken from the official NIST TREC 2002 evaluation of those systems. Two additional rows are provided on TREC-02 benchmark that compare our results to the best and average systems, respectively, among the 54 SBD runs submitted by TREC participants.

As anticipated, the SBD performance on TREC-02 data was lower than on TREC-01 data set. This was very noticeable in other participating systems as well. Never-the-less, the error rates of the

Sys.	Video Data	All		Cuts		Gradual		Frame	
		Rc	Pr	Rc	Pr	Rc	Pr	Rc	Pr
alm1	TR-01	.95	.88	.98	.97	.87	.68	.59	.93
sys47	TR-01	.96	.92	.99	.98	.89	.79	.66	.90
alm1	TR-02	.86	.77	.93	.80	.69	.71	.48	.94
sys47	TR-02	.88	.83	.93	.87	.76	.72	.57	.89
S-5	TR-02	.84	.89	.91	.94	.76	.78	.62	.90
mean	TR-02	.76	.79	.86	.84	.53	.60	.55	.71

Table 1: Shot boundary detection results, comparing the new system with last year system on both TREC-01 and TREC-02 video data test sets. If all participating systems are to be ranked by $Pr_{All} + Rc_{All}$ then system *S-5* would be found the best one, provided here for comparison. System *mean* reflects the average of all 54 submitted systems.

new system *sys47* were $20-36\%$ lower than of the baseline system *alm1* in almost all measures on both data sets.

4.2 Concept detection results

Overall, concept detection results were submitted for ten concept classes. The evaluation results are plotted in Figure 4, which shows shows Average Precision measured at a fixed number of documents (1000 for the feature test set). The "Average" bars correspond to the performance averaged across all participants. The "Best" bars correspond to the system returning the highest Average Precision. The "IBM" bars correspond to IBM's submitted concept detection run (priority=1). The IBM system performed relatively well on the concept detection task giving highest Average Precision on 6 of the 10 concepts[7].

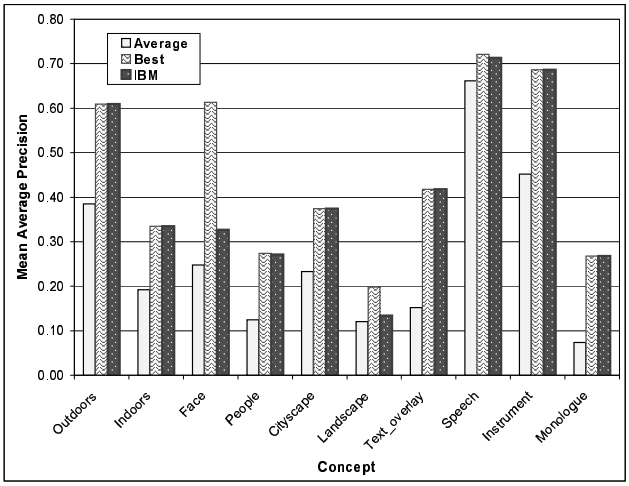

Figure 4: Comparison of concept detection performance using Average Precision.

[7]Top score is indicated only on five concepts. In our original submission to NIST, we mistakenly submitted the speech detection twice overwriting our instrument detection result. However, the actual Average Precision of our instrument sound detector was 0.686, which was reported through later communication with NIST.

4.3 Concept exchange results

Apart from running the primary and secondary detectors on the search test set to assist the search task, we participated in the concept exchange task by submitting results of eight primary detectors on the search test set. We generated shot based MPEG-7 descriptions for this exercise thus permitting easy exchange of the detection results between participants.

4.4 Search results

The search task required retrieving video shots from the search test collection for a given set of query topics. We investigated both manual and interactive methods of searching. We submitted four runs of all 25 query topics using the content-based, model-based, speech-based, and interactive search methods described above. Table 2 summarizes the results for the four search runs.

System	Type	Code	MAP
CBR	Manual	M_B_M_1	0.006
SDR	Manual	M_B_M-2_2	0.137
CBR+SDR	Manual	M_B_M-3_3	0.093
CBR+SDR	Interactive	I_B_M-4_4	0.244

Table 2: Summary of search results for four submitted runs.

4.4.1 Manual CBR

The manual CBR run consisted of mapping the query topics into one or more content-based or model-based queries and fusing the results in a predetermined fashion. As described in Section 2.2, CBR was based on a variety of descriptors. The manual CBR run was generated by allowing the following operations to answer each query topic:

1. Issue a content-based search by selecting one or more query examples, a feature type, and a fusion method, as necessary;
2. Issue a model-based search by selecting one or more concept models, a fusion method, and model weights, as necessary;
3. Fuse results lists from one or more content-based or model-based search by selecting a fusion method.

For example, the following sequence of operations was executed for Query 79: *People spending leisurely time at the beach*:

1. Pick examples 0, 1, 4, 8, 12, 23, 29 from query content set
2. Perform CBR search with edge histogram layout using "minimum" fusion (Eq 15)
3. Combine with "Landscape" model using "intersection" combining method (Eq 2) and "product" aggregation function (Eq 8).

The exact mapping of query topics into a fixed sequence of the above operations was performed manually by visually optimizing performance over the FTR and/or FV collections without knowledge of the search test collection. Once a query topic was mapped to system operations, the operations were applied to the search collection by a designated person who did not participate in the mapping process or have prior knowledge of the search test collection. Figure 5 shows the results for topic 76, which is looking for shots depicting "James Chandler." As shown, some matches

Figure 5: Results for topic 76: James Chandler.

are found in the results list, however, many shots of "James Chandler" are not retrieved using CBR.

With respect to performance, it was our experience that the TREC 2002 query topics were at a higher semantic level than what CBR can handle. While CBR and semantic modeling are generally able to capture low- to mid-level semantics, they are fairly limited in the case of only a few query examples or mid- to high-level semantics. We found that purely CBR worked best for refining candidate lists generated from semantically rich sources, such as speech, or explicit semantic models that closely match the query need. For example, refining the face model by cross-comparison with examples images of "James Chandler" did produce a few relevant hits near the top (see Figure 5). Model-based retrieval on the other hand worked well when the query topic was a close match to an existing model and was built with sufficient training data, such as the "musician" topic. However, in the case of limited example content, such as of query topic looking for "butterflies", or given a lack of closely related explicit semantic models, CBR and MBR techniques alone are not sufficient. In addition, some of the query topics were so general (e.g., beach query) or specific (e.g., Price Tower query) that it is doubtful whether any reasonable discrimination can be done using low-level features alone.

4.4.2 Manual SDR

Manual searching using spoken document retrieval (SDR) was based on the indexed speech information. We explored multiple methods of SDR and their fusion, where the SDR queries were developed through interaction with the Feature Training collection.

Query strings were created manually for each query. Queries derived from the audio and textual statement of information need supplied by NIST were expanded by hand in ad-hoc fashion based on retrieval on the FTR+FV sets[8]. More complicated query strings are used in the Boolean system, since it was hypothesized that the Boolean retrieval would be less susceptible to the effects of query over-tuning on FTR+FV.

The query terms used in the submitted multiple-SDR fusion system for topic 90 ("Find shots with one or more snow-covered mountain peaks or ridges. Some sky must be visible behind them") were "ice snow covered mountain peaks valley vista". Twenty relevant items were retrieved in the top 100, with Average Precision 0.12. For topic 84 ("Find shots of Price Tower, designed by Frank Lloyd Wright and built in Bartlesville, Oklahoma") the query terms are "Price Tower Frank Lloyd Wright Bartlesville Oklahoma", the top three items recalled are relevant and Average Precision is 0.75.

Weights for the SDR-FUSION-SYSTEM were optimized using the limited ground truth that was compiled for FTR+FV. As expected, this scheme led to Mean Average Precision (MAP) improvements FTR+FV; more importantly, fusion gave performance improvements (35%) over our best single SDR system on the unseen search test data (as shown in Table 3). Note that simple post-evaluation changes in the video segment-to-shot mapping scheme improved the performance of the individual OKAPI systems (eg. OKAPI-SYSTEM-1 increased to MAP 0.114) and the fusion system performance might be expected to improve further as the component systems improve. The results overall are a significant improvement over those for IBM's speech-only retrieval submission to TREC-2001. The system was ranked second among 27 evaluated manual search results.

System	MAP
OKAPI-SYSTEM-1	0.073
OKAPI-SYSTEM-2	0.093
BOOLEAN-SYSTEM-1	0.101
SDR-FUSION-SYSTEM	0.137

Table 3: Search test performance of the fusion system and its three components.

4.4.3 Manual CBR and SDR

The combination of CBR and SDR was explored for manual searching, where queries were developed through interaction with the Feature Training collection. An example of (successful) SDR and CBR integration is query topic 86 ("find overhead views of cities - downtown and suburbs; the viewpoint should be higher than the highest building visible"). In the following, we assume that the SDR results and CBR results have been found independently prior to the integrated query:

1. Retrieve results for SDR query of "view panorama overhead downtown suburbs city town urban"
2. Expand results list to include adjacent shots (repeat two times) using expand operation (see Section 3.8)
3. Combine with CBR results using "union" combination method (Eq 3) and "product" aggregation function (Eq 8).

The final Average Precision improved from CBR 0.0 and SDR 0.039 to CBR+SDR 0.057. A similar approach was used for the other queries with minor differences such as the number of shot expansions and the choice of the combination method and aggregation function, for example, using "intersection" rather than "union" and "sum" rather than "product". However, this approach

[8]Later experiments showed that, at least in OKAPI-SYSTEM-1, the gains due to the manual query expansion were negligible.

was not always successful; for example, the same scheme was used for topic 84 ("Price Tower") SDR+CBR but performance was degraded below that obtained using SDR alone. This approach to SDR and CBR integration improved 4 of the 25 queries beyond the performance attained with SDR alone.

4.4.4 Interactive search

We explored interactive search using CBR and SDR in which the user interacted with the search test collection at query-time, we chose various combinations of these methods and selected among different methods for fusion, multiple examples search, relevance feedback, and browsing. The wall-clock time was measured to gauge the user effort for each interactive query. The following describes the interactive search operations for query topic 89 for "Butterflies", which took just over seven minutes of user time:

1. Search for shots of butterflies using SDR with terms such as "monarch", "butterfly", "wings", "flower".
2. View grouping of results by video (clusters shots according to source video) to get idea of which videos contribute which shots
3. Remove two irrelevant shots at top of results list
4. Expand all shots t adjacent shots
5. Results show 5 hits at the top, stop.

5 Summary

We presented the IBM Research video indexing system. The system explores fully-automatic content analysis methods for shot detection, multi-modal feature extraction, statistical modeling for semantic concept detection, and speech recognition and indexing. The system supports manual methods of querying based on automatically extracted features, models, and speech information. In this paper we described the system and the experiments runs that are part of the TREC-2002 video retrieval benchmarking effort. The results show good performance on tasks such as shot boundary detection, concept detection, and search.

Acknowledgments: We thank Prof. Chiou-Ting Hsu, National Tsing-Hua University, Hsinchu, Taiwan and her students for their assistance in annotating the feature training data sets.

References

[1] P. Salembier and J. R. Smith. MPEG-7 multimedia description schemes. *IEEE Trans. Circuits Syst. for Video Technol.*, August 2001.

[2] *IBM CueVideo Toolkit Version 2.1, http://www.almaden.ibm.com/cs/cuevideo/*.

[3] J. R. Smith, S. Srinivasan, A. Amir, S. Basu, G. Iyengar, C.-Y. Lin, M. Naphade, D. Ponceleon, and B. Tseng. Integrating features, models, and semantics for trec video retrieval. In E. M. Voorhees and D. K. Harman, editors, *Proc. Text Retrieval Conference (TREC)*, pages 240–249, Gaithersburg, MD, 2002. NIST.

[4] S. Srinivasan, D. Ponceleon, A. Amir, , and D. Petkovic. What is in that video anyway? in search of better browsing. In *in Proc. of IEEE Intl. Conf. on Multimedia Computing and Systems*, pages 388–392, Florence, Italy, 1999.

[5] J. R. Smith and A. Natsev. Feature and spatial normalization for content-based retrieval. In *IEEE Conference on Multimedia and Expo*, Laussane, Switzerland, August 2002.

[6] J. R. Smith. Content-based access of image and video libraries. In A. Kent, editor, *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc., 2001.

[7] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Spatial color indexing and applications. *International Journal of Computer Vision*, 35(3):245–268, December 1999.

[8] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Trans. Syst., Man, Cybern.*, SMC-8(6):460–473, 1978.

[9] R. Jain, R. Kasturi, and B. Schunck. *Machine Vision*. MIT Press and McGraw-Hill, New York, 1995.

[10] M. Naphade, S. Basu, J. Smith, C. Lin, and B. Tseng. Modeling semnatic concepts to support query by keywords in video. In *IEEE International Confernce on Image Processing*, Rochester, NY, Sep 2002.

[11] M. Naphade, T. Kristjansson, B. Frey, and T. S. Huang. Probabilistic multimedia objects (multijects): A novel approach to indexing and retrieval in multimedia systems. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 536–540, Chicago, IL, Oct. 1998.

[12] M. R. Naphade, I. Kozintsev, and T. S. Huang. A factor graph framework for semantic video indexing. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(1):40–52, Jan 2002.

[13] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[14] G. Iyengar, H. Nock, and C. Neti. Audio-visual synchrony for detection of monologues in video archives. In *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, April 2003.

[15] J.-C. Shim, C. Dorai, and R. Bolle. Automatic text extraction from video for content-based annotation and retrieval. In *IEEE Conference on Pattern Recognition*, volume 1, pages 618–620, Brisbane, Australia, August 1998.

[16] A. Jaimes, M. Naphade, H. Nock, J. R. Smith, and B. Tseng. Context-enhanced video understanding. In *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases 2003*, San Jose, CA, January 2003.

[17] A. Amir, A. Efrat, and S. Srinivasan. Advances in phonetic word spotting. In *Proc. of the 2001 ACM CIKM Int. Conference on Information and Knowledge Management*, pages 580–582. ACM, November 2001.

[18] S. E. Robertson, A. Walker, K. Sparck-Jones, M. M. Hancock-Beaulieu, and M. Gatford. OKAPI at TREC-3. In *In Proc. Third Text Retrieval Conference*, 1995.

[19] J. Xu and B. Croft. Improving the Effectiveness of Informational Retrieval with Local Context Analysis. *ACM Transactions on Information Systems*, 2000.

[20] T. V. Ashwin, R. Gupta, and S. Ghosal. Leveraging non-relevant images to enhance image retrieval performance. In *Proc. ACM Intern. Conf. Multimedia (ACMMM)*, Juan Les Pins, France, December 2002.