Integrating Features, Models, and Semantics for TREC Video Retrieval

John R. Smith†, Savitha Srinivasan‡, Arnon Amir‡, Sankar Basu†, Giri Iyengar†,
Ching-Yung Lin†, Milind Naphade†, Dulce Ponceleon‡, Belle Tseng†
†IBM T. J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532 USA
‡IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120 USA

Abstract

In this paper, we describe a system for automatic and interactive content-based retrieval of video that integrates features, models, and semantics. The novelty of the approach lies in the (1) semi-automatic construction of models of scenes, events, and objects from feature descriptors, and (2) integration of content-based and model-based querying in the search process. We describe several approaches for integration including iterative filtering, score aggregation, and relevance feedback searching. We describe our effort of applying the content-based retrieval system to the TREC video retrieval benchmark.

1 Introduction

The growing amounts of digital video are driving the need for more effective methods for storing, searching, and retrieving video based on its content. Recent advances in content analysis, automatic feature extraction, and classification are improving capabilities for effectively searching and filtering digital video using information based on perceptual features, content structure, models, and semantics. The emerging MPEG-7 multimedia content description standard promises to further improve content-based searching by providing a rich set of standardized tools for describing multimedia content in XML [SS01]. However, MPEG-7 does not standardize methods for extracting descriptions nor for matching and searching. The extraction and use of MPEG-7 descriptions remains a challenge for future research, innovation, and industry competition [Smi01].

In this paper, we describe a system for automatic and interactive content-based retrieval that integrates features, models, and semantics [SBL$^+$01]. The system analyzes the video by segmenting it into shots, selecting key-frames, and extracting audio-visual descriptors from the shots. This allows the video to be searched at the shot-level using content-based retrieval approaches. However, we further analyze the video by developing and applying models for classifying content. The approach requires the manual- or semi-automatic annotation of the video shots to provide training data. The models are subsequently used to automatically assign semantic labels to the video shots. In order to apply a small number of models but have at the same time to have large impact on classifying the video shots, we have primarily investigated models that apply broadly to video content, such as indoor *vs.* outdoor, nature *vs.* man-made, face detection, sky, land, water, and greenery. However, we have also investigated several specific models including airplanes, rockets, fire, and boats. While the models allow the video content to be annotated automatically using this small vocabulary, the integration of the different search methods together (content-based and model-based) allows more effective retrieval.

In the paper, we describe the approach for integrating features, models, and semantics in a system for content-based retrieval of video. We have applied these systems and methods to the NIST TREC video retrieval benchmark, which consists of 74 queries of a video corpus containing approximately 11 hours of video. The queries, which were designed to access video based on semantic contents, permit automatic and/or interactive approaches for retrieving the results. We enhance the automatic retrieval by using the models in conjunction with the features to match the query content with the target video shots. For interactive retrieval, we allow the user to apply several methods of iterative searching that combines features, semantics, and models using different filtering operations and weighting methods. In this paper, we describe more details about the approach and discuss results for the TREC video retrieval benchmark.

2 Content analysis system

The video content is analyzed through several processes that involve shot detection, feature extraction, and classification, as shown in Figure 1. The video is segmented temporally according to shot boundaries, and descriptors are extracted for each shot. The descriptors are ingested into a storage system. The descriptors are used as input into the model-based classification system which assigns semantic labels to each shot. The system also ingests any meta-data related to the content such as title, format, source, and so forth.

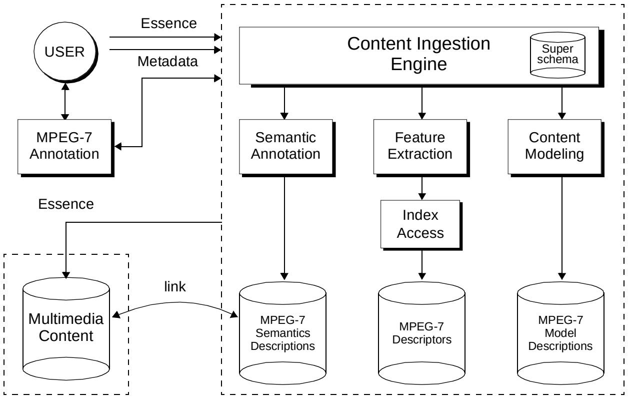


Figure 1: The video content ingestion engine first segments the video temporally using shot detection and selects key-frames, then extracts descriptors of the audio-visual features and applies models in order to classify the content.

2.1 Shot detection

The video content is pre-processed by splitting it into temporal segments using the *IBM CueVideo* (program `cuts.exe` with the default settings) [Cue]. After the shots are detected, key-frames are selected and extracted, and all MPEG I-frames are extracted, as shown in Figure 2. These images are stored and indexed and are used for accessing the shots.

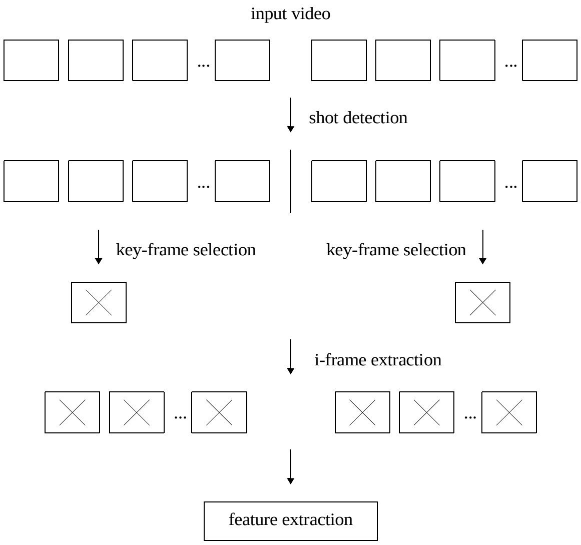


Figure 2: The shot detection system automatically segments the video into temporal segments and selects a key-frame for each shot.

CueVideo 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 frames. Frame pairs at one, three and seven frames apart and their corresponding thresholds are shown by the three upper graphs in Figure 3. Statistics of frame differences are computed in a moving window around the processed frame and are used to compute the adaptive thresholds. Hence the program does not require sensitivity-tuning parameters.

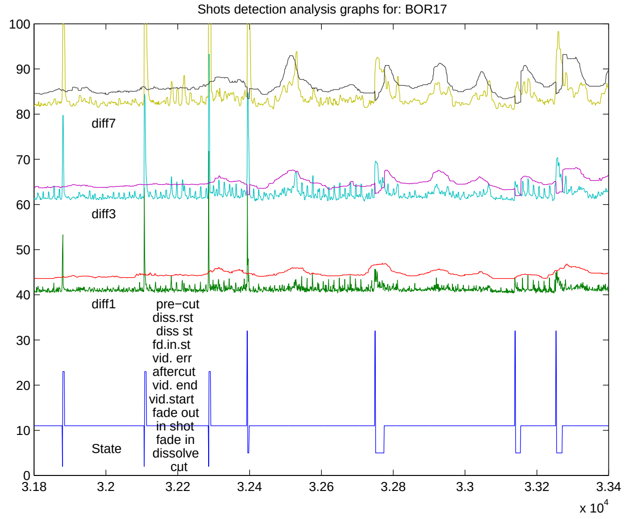


Figure 3: This example represents a 53 Seconds sequence with four cuts and three dissolves in high noise (from *bor17.mpg*, frame numbers: 31800–33400). The middle cut is mistakenly detected as a short dissolve (Alm2).

A state machine is used to detect and classify the different shot boundaries, shown at the botom of Figure 3 with all thirteen states listed. At each frame a state transition is made from the current state to the next state, and any required operation is taken (e.g., report a shot, save a key-frame to file). The algorithm classifies shot boundaries into Cuts, Fade-in, Fade-out, Dissolve and Other. It works in a single pass, is robust to possibly uncompliant MPEG streams, and runs about 2X real time on a 800MHz P-III.

2.2 Feature extraction

The system extracts several different descriptors for each of the key-frames and i-frames. We have used the following descriptors:

1. color histogram (166-bin HSV color-space),
2. grid-based color histogram (4x4 grid of the HSV histogram),
3. texture spatial-frequency energy (variance measure of

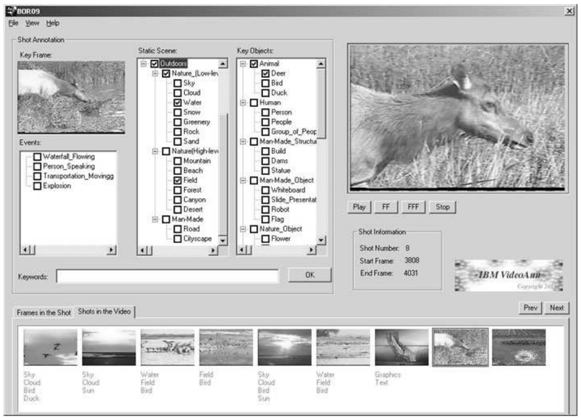
each of 12 bands of quadrature mirror filter wavelet decomposition, and

4. edge histogram (using Sobel filter and quantization to 8 angles and 8 magnitudes).

Each of these descriptors is stored and indexed separately. However, at retrieval time, the CBR matching function allows the descriptor values to be combined using an arbitrary weighting function in order to determine the similarity of the query and target images based on multiple features.

2.3 Semi-automatic annotation

In order to allow a model-based approach to video retrieval, ground-truth data is needed for training the models. In order to create training data, we developed a video annotation tool that allows the users to annotate each shot in the video sequence, as shown in Figure 4. The tool allows the user to identify and label scenes, events, and object by applying the labels at the shot-level. The tool also allows the user to associate object-labels with individual regions in a key-frame.



Events:
- □ Waterfall_Flowing
- □ Person_Speaking
- □ Transportation_Movingg
- □ Explosion

Static Scene:
- ☑ Outdoors
  - ☑ Nature_(Low-lev
    - □ Sky
    - □ Cloud
    - ☑ Water
    - □ Snow
    - □ Greenery
    - □ Rock
    - □ Sand
  - □ Nature(High-lev
    - □ Mountain
    - □ Beach
    - ☑ Field
    - □ Forest
    - □ Canyon
    - □ Desert
  - □ Man-Made
    - □ Road
    - □ Cityscape

Key Objects:
- ☑ Animal
  - ☑ Deer
  - □ Bird
  - □ Duck
- □ Human
  - □ Person
  - □ People
  - □ Group_of_Peop
- □ Man-Made_Structu
  - □ Build
  - □ Dams
  - □ Statue
- □ Man-Made_Object
  - □ Whiteboard
  - □ Slide_Presentat
  - □ Robot
  - □ Flag
- □ Nature_Object
  - □ Flower

Keywords:

Shot Information
Shot Number: 8
Start Frame: 3808
End Frame: 4031

Sky
Cloud
Bird
Duck — Sky
Cloud
Sun — Water
Field
Bird — Field
Bird — Sky
Cloud
Bird
Sun — Water
Field
Bird — Graphics
Text

Figure 4: The video annotation tool allows users to label the events, scenes, and objects in the video shots.

For annotating video content, we created a lexicon for describing events, scenes, and objects; the following excerpt gives some of the annotation terms:

- **Events**: water skiing, boat sailing, person speaking, landing, take-off/launch, and explosion;
- **Scenes**: outer space (moon, mars), indoors (classroom, meeting room, laboratory, factory), outdoors (nature, sky, clouds, water, snow, greenery, rocks, land, mountain, beach, field, forest, canyon, desert, waterfall), and man-made (road, cityscape);
- **Objects**: non-rigid objects (animal, deer, bird, duck, human), rigid objects (man-made structure, building, dam, statue, tree, flower), transportation (rocket, space shuttle, vehicle, car, truck, rover, tractor), and astronomy.

The video anntotation tool allows the user to process the video shot-by-shot, and assign the labels to each shot. The tool is semi-automatic in that it automatically propagates labels to “similar” shots as described in [NLS$^+$02]. The system requires the user to confirm or reject the propagated labels.

2.4 Content modeling

The content modeling system uses the labeled training video content to classify other video content (in our case, the test TREC video corpus). We have investigated several different types of static models including Bayes nets, multinets [NKHR00], and Gaussian mixture models. In some cases, we have used additional descriptors in the models, which are not applied for content-based retrieval, such as motion activity and color moments.

We have developed statistical models for the following concepts:

- **Events**: fire, smoke, launch;
- **Scenes**: greenery, land, outdoors, rock, sand, sky, water;
- **Objects**: airplane, boat, rocket, vehicle.

2.4.1 Statistical modeling

In the statistical modeling approach, the descriptors extracted from the video content are modeled by a multidimensional random variable X. 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. In particular, each semantic concept is represented by a binary random variable. The two hypotheses associated with each such variable are denoted by H_i, $i \in \{0,1\}$, where 0 denotes absence and 1 denotes presence of the concept. Under each hypothesis, we assume that the descriptor values are generated by the conditional probability density function $P_i(X)$, $i \in \{0,1\}$.

In case of scenes, we use static descriptors that represent the features of each key-frame. In case of events, which have temporal characteristics, we construct temporal descriptors using time series of static descriptors over the multiple video frames. We use a *one-zero* loss function [Poo99] to penalize incorrect detection. This is shown in Equation 1:

$$\lambda(\alpha_i|\omega_j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

The risk corresponding to this loss function is equal to the average probability of error and the conditional risk with action α_i is $1 - P(\omega_i|x)$. To minimize the average probability of error, class ω_i must be chosen, which corresponds to the maximum a posteriori probability $P(\omega_i|x)$. This corresponds to the minimum probability of error (MPE) rule.

In the special case of binary classification, the MPE rule can be expressed as deciding in favor of ω_1 if

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{(\lambda_{12} - \lambda_{22})P(\omega_2)}{(\lambda_{21} - \lambda_{11})P(\omega_1)} \tag{2}$$

The term $p(x|\omega_j)$ is the *likelihood* of ω_j and the test based on the ratio in Equation (2) is called the *likelihood ratio test* (LRT) [DH73, Poo99].

2.4.2 Parameter estimation

For modeling the TREC video content, we assume that the conditional distributions over the descriptors X under the two hypotheses – concept present (H_1) and concept absent (H_0) – have been generated by distinct mixtures of diagonal Gaussians. The modeling of these semantic concepts involves the estimation of the unknown but determinsitic parameters of these Gaussian mixture models (GMMs) using the set of annotated examples in the training set. For this purpose the descriptors associated with training data corresponding to each label are modeled by a mixture of five gaussians. The parameters (mean, covariance, and mixture weights) are estimated by using the Expectation Maximization (EM) [DLR77] algorithm.

The rest of the training data is used to build a negative model for each label in a similar way, which corresponds to a garbage model for that label. The LRT is used in each test case to determine which of the two hypotheses is more likely to account for the descriptor values. The likelihood ratio can also be looked upon as a measure of the *confidence* of classifying a test image to the labeled class under consideration. A ranked list of confidence measures for each of the labels can be produced by repeating this procedure for all the labels under consideration.

2.4.3 Region merging

We use manually assigned bounding boxes encompassing regions of interest obtained during annotations for extracting features. The testing is also done at the regional bounding box level. To fuse decisions from several bounding boxes in a key-frame, we use the following hypothesis: If a concept is to be declared absent in a frame, it must be absent in each and every bounding box tested. We can then compute the product of the probability of the ”concept absent” hypothesis to obtain the probability of the concept being absent in the frame. Alternately, we can also use the maximum possible probability of the concept being detected in any region as the probability of its occurrence in the image/frame. For concepts which are global in terms of feature support, this step is not needed. Localized or regional concepts include *rocket*, *face*, *sky*, and so forth.

2.4.4 Feature fusion

The objective of feature fusion is to combine multiple statistical models for the different video features. Separate GMM models are used for each of the different descriptors (e.g., color histogram, edge direction histogram, texture, and so forth). This results in separate classifications and associated confidence for each test image depending on the descriptor. While the classifiers can be combined in a many ways, we explored straightforward methods such as taking sum, maximum,and product of the individual confidences for each descriptor in computing an overall classification confidence.

While this strategy of “late feature fusion” is fairly simple, one can envision other “early feature fusion” methods such as concatenating different descriptors into a single vector and then building a single GMM. We did not pursue this strategy due to the large dimensionality of the descriptors, especially in view of the paucity of training video content depicting the concepts of interest. However, it may be possible to consider discrimination in reduced dimensional subspaces of the feature space by using techniques such as the principal component analysis (PCA) or by using more sophisticated dimensionality reduction techniques that would allow concatenation and modeling of high-dimensional descriptors.

2.4.5 Training

The performance of statistical models such as the GMM depend to a large extent on the amount of training data. Due to the relatively small amount of labeled training video data beyond the TREC video corpus, we adopted a “leave one clip out strategy.” This means that we trained a model for each concept as many number of times as the number of video clips. During each such training, one clip was left out from the training set. The models for the two hypotheses thus trained were used to detect the semantic concept in the clip that was left out.

2.5 Speech indexing

In addition to automatic analysis and modeling of the features of the video content, we also investigated the use of speech indexing as an alternative approach for video retrieval [PS01]. We used the *IBM ViaVoice* speech recognition engine to transcribe the audio and generate a continuous stream of words. We define a unit-document to be a 100 word temporal segment where consecutive segments overlap partially in order to address the boundary truncation effect. There are several operations performed in sequence in this processing.

First, the words and times from the recognizer output are extracted to create the unit-document files with associated timestamps. The Julian time at the start of the audio is used as the reference basis. This is followed by tokenization to detect sentence/phrase boundaries and then part-of-speech tagging such as noun phrase, plural noun etc. The morphological analysis uses the part-of-speech tag and a morph dictionary to reduce each word to its morph. For example, the verbs, lands, landing and land will all be reduced to land. Then, the stop words are removed using a standard stop-words list. For each of the remaining words, the number of unit-documents that it belongs to (the inverse document frequency) is computed and is used to weight these word.

3 Video retrieval

Once the video content is ingested, the descriptors and model results are stored and indexed. This allows the user to carry out the searches in a video query pipeline process as shown in Figure 6, in which queries are processed in a multi-stage search in which the user selects models and clusters or examples of video content at each stage. By operating on the interim results, the user controls the query refinement. As shown in Figure 6, at each stage of the search, a query Q_i produces a result list R_i. The result list R_i is then used as input into a subsequent query Q_{i+1}, and through various selectable operations for combining and scoring R_i with the matches for Q_{i+1}, the result list R_{i+1} is produced. The user can continue this iterative search process until the desired video content is retrieved.

3.1 Content-based retrieval

Content-based retrieval is the most amenable to automatic retrieval in the case that the query provides example content. For TREC video retrieval, each of the queries provided example content which included anywhere from a single image to several video clips. For automatic content-based retrieval, the following approach was adopted: the query content was analyzed using shot detection, key-frame selection, and feature extraction to produce a set of descriptors of the query content. Then, the query descriptors were matched against the target descriptors. We considered two approaches for automatic content-based matching: (1) matching of descriptors of the query and target key-frames, and (2) matching of descriptors for multiple frames (i-frames) from the query and target video, as shown in Figure 5.

3.1.1 Multi-frame matching

For multi-frame matching, different semantics of the matching are possible depending on the nature of the query. For example, if all of the individual images in the query content are important for the query (“all” semantics), then the matching semantics is such that the best target video shot from the database should have the best overall score of matching all of the query images to images in the target shot.

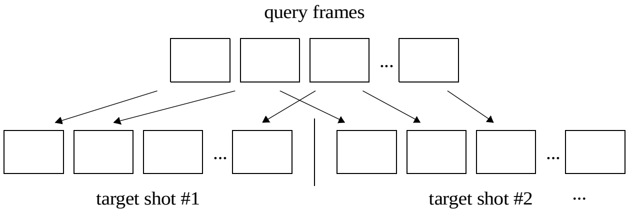


Figure 5: Content-based retrieval matches multiple query frames against multiple frames in the target shots.

Multi-frame matching requires first the determination of the best matches among individual images from the query and target, and then computation of the overall score of all the matches. However, alternatively, if the query images are meant to illustrate different variations of the content (“or” semantics), then the matching semantics is such that the best target video should be the ones that have a single frame that best matches one of the query images.

3.1.2 Interactive retrieval

For interactive retrieval, we enhanced the content-based approach by allowing the user to conduct multiple rounds of searching operations in which each successive round refines or builds on the results of a previous round. Each round consists of the following: (1) a similarity search in which target shots are scored against query content (using single frame or multi-frame search), and (2) a combining of these search results with the previous results list. This way, each successive round combines new results with a current list. We investigated several ways of combining results which involve different ways of manipulating the scores from the successive rounds. We have used a choice of the following aggregation functions for combining the scores:

$$D_i(n) = D_{i-1}(n) + D_q(n), \tag{3}$$

and

$$D_i(n) = \min(D_{i-1}(n), D_q(n)), \tag{4}$$

where $D_q(n)$ gives the score of video shot n for the present query, and $D_{i-1}(n)$ gives the combined score of video shot n for the previous query, and $D_i(n)$ gives the combined score result for the current round. Eq. 3 simply takes the sum of the score of each target video shot for the current

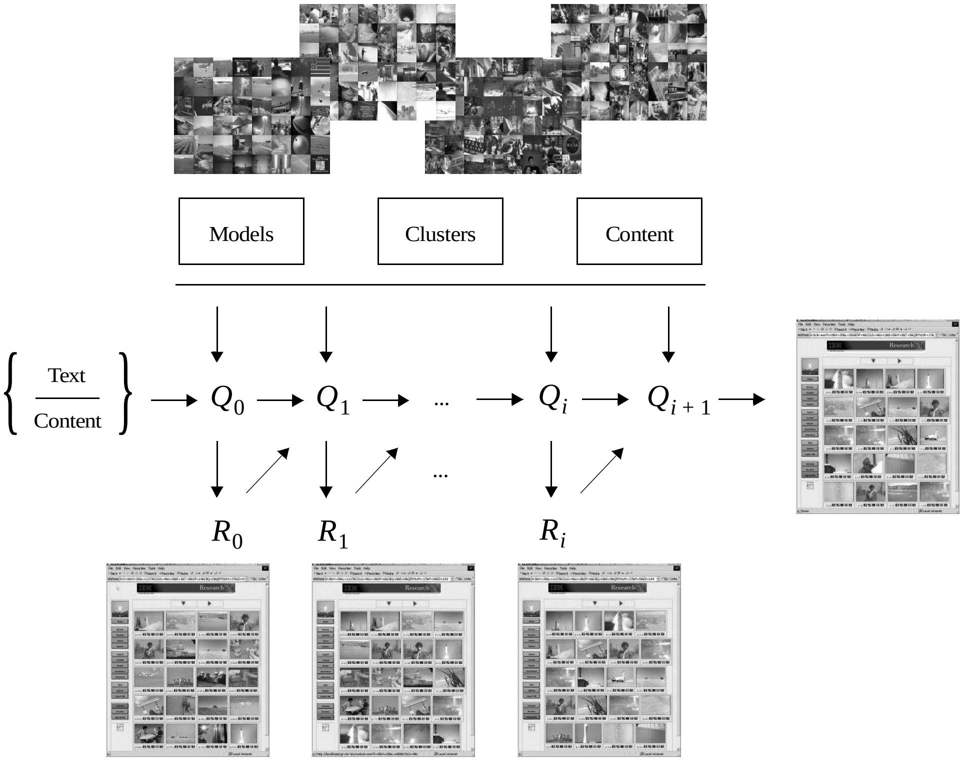


Figure 6: The video content retrieval engine integrates methods for searching in an iterative process in which the user successively applies content-based and model-based searches.

query plus the cumulative score of the previous queries. This has the effect of weighting the most recent query equally with the previous queries. Eq. 4 takes the minimum of the current score and the previous scores for each target video shot. This has the effect of ranking most highly the target shots that best match any one of the query images. Although, Eq. 3 and Eq. 4 are simple monotonic functions, other combining functions that use arbitrary join predicates are possible [NCS$^+$01].

For combining content-based and model-based retrieval, we allow the above methods for combining results, however, we allow additionally a filtering method that computes the intersection of the previous result list with the results from the current query, as described next.

3.2 Model-based retrieval

The model-based retrieval allows the user to retrieve the target shots based on the semantic labels produced by the models. Each semantic label has an associated confidence score. The user can retrieve results for a model by issuing a query for a particular semantic label. The target video shots are then ranked by confidence score (higher score gives lower rank). Since the models do not assign labels to all of the target shots, only the ones that are positively classified tothe semantic class, the model-based search does not give a total ranking of the target shots. That is, the model-based search both filters and ranks the target shots, which has implications for its use in iterative searching.

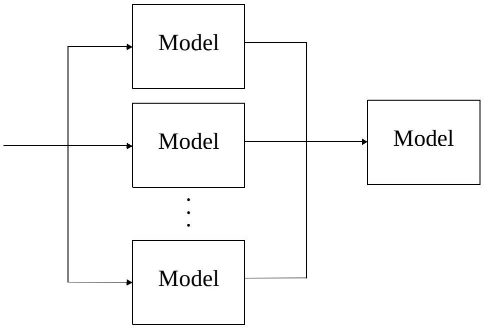


Figure 7: Parallel model search allows the user to define weighting of multiple models.

The models can be applied sequentially or in parallel as shown in Figure 7. In the case of parallel search, the user defines weighting of multiple models in a single query. In sequential search, the user decides based on interim results which models to apply. For example, a parallel model-based search is as follows: nature $= 0.5*$outdoors$+0.25*$water$+$

$0.25 * \text{sky}$. An example sequential model-based search is as follows: outdoors → no faces → no water.

3.3 Video query pipeline

The integrated search is carried out by the user successively applying the content-based and model-based search methods as shown in Figure 8.

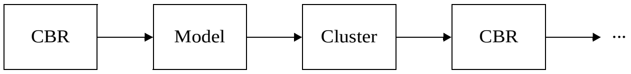


Figure 8: Integration of content-based and model-based searching in the video query pipeline.

For example, a user looking for video shots showing a beach scene can issue the following sequence of queries in the case that beach scenes have not been explicitly labeled::

1. Search for model = "outdoors",
2. Aggregate with model = "sky",
3. Aggregate with query image (possibly selected image) resembling desired video shot,
4. Aggregate with model = "water",
5. Aggregate with selected relevant image, video shot,
6. Repeat.

The iterative searching allows the users to apply sequentially the content-based and model-based searches. Different options can be used for scoring the results at each stage of the query and combining with the previous results. For TREC video retrieval, a choice of the following different approaches using different aggregation functions were provided for combining the scores:

1. **Inclusive**: each successive search operation issues new query against target database:
$$D_0(n) = D_q(n), \tag{5}$$
2. **Iterative**: each successive search operation issues query against current results list and scores by new query:
$$D_i(n) = D_q(n), \tag{6}$$
3. **Aggregative**: each successive search operation issues query against current results list and aggregates scores from current results and new query results:
$$D_i(n) = f(D_{i-1}(n), D_q(n)), \tag{7}$$

where $f(.)$ corresponds to $\min$, $\max$, or avg. The distance scores $D_i(n)$ are based on feature similarity (for CBR) and label confidence (for models). For the models, $D_q(n) = 1 - C_q(n)$, where $C_q(n)$ gives the confidence of the query label for video shot n, and $D_{i-1}(n)$, and $D_i(n)$ are defined as above. The lossy filtering is accounted for in that some target shots n^* have confidence score $C_q(n^*) = -\infty$. Eq. 7 combines the label score of each target video shot for the current query plus the cumulative label score of the previous queries, whereas Eq. 6 takes only the latest score.

3.4 Speech retrieval

To compute the video retrieval results using speech indexing for the TREC video retrieval, we used the textual statement of information need associated with each topic without any refinement or pruning of the text. The speech retrieval system works as follows: the system first loads the inverted index and precomputed weights of each of the non-stop words. A single pass approach is used to compute a relevancy score with which each document is ranked against a query, where the relevancy score is given by the Okapi formula [RWSJ$^+$95].

Each word in the query string is tokenized, tagged, morphed and then scored using the Okapi formula above. The total relevancy score for the query string is the combined score of each of the query words. The scoring function takes into account the number of times each query term occurs in the document normalized with respect to the length of the document. This normalization removes bias that generally favor longer documents since longer documents are more likely to have more instances of any given word.

4 Retrieval system

We have applied this type of iterative and integrated content-based and model-based searching procedure for computing the results for many of the TREC video retrieval topics. Example topics for which this approach was used include: "scenes with sailing boats on a beach,", "scenes with views of canyons," and "scenes showing astronaut driving a lunar rover." The video retrieval system is illustrated in Figure 9.

4.1 Benchmark

The TREC video retrieval benchmark[1] was developed by NIST[2] to promote progress in content-based retrieval (CBR) from digital video via open, metrics-based evaluation. The benchmark involves the following tasks:

- Shot boundary detection

[1] http://www-nlpir.nist.gov/projects/t01v/revised.html

[2] http://trec.nist.gov/call01.html

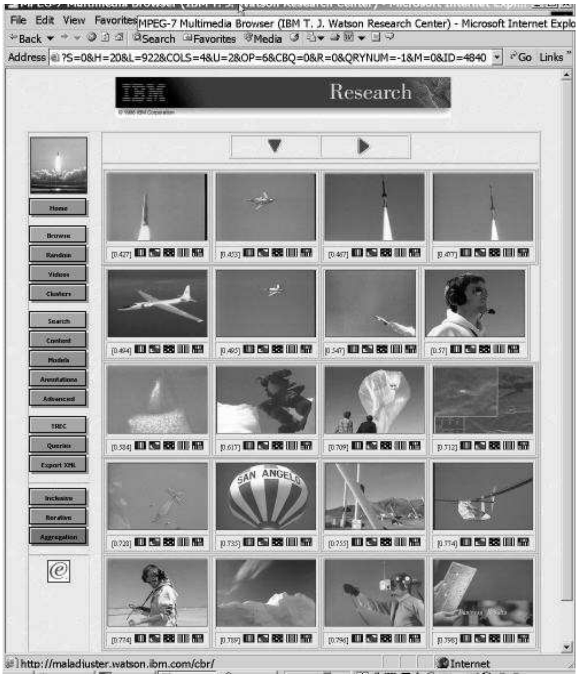


Figure 9: Screen image of the video retrieval system.

- Known item search

- General statements of information need.

The benchmark consists of the following:

- Approximately 11 hours of video

- 74 query topics, which include statements of information needs in text and example content

- Ground truth assessments (provided by participants for known-item queries)

- Quantitative metrics for evaluating retrieval effectiveness (i.e., precision *vs.* recall).

The benchmark focuses on content-based searching in that the use of speech recognition and transcripts is not emphasized. However, the queries themselves typically involve information at the semantic-level, i.e., "retrieve video clips of Ronald Reagan speaking," and opposed to "retrieve video clips that have this color." The two kinds of queries, known-item and general information need, are distinguished in that the number of matches for the known-item queries is pre-determined, i.e., it is known that there are only two clips showing Ronald Reagan. On the other hand, for the general searches, the number of matches in the corpus in not known, i.e., "video clips showing nature scenes."

4.2 Shot detection benchmark results

The results of the shot boundary detection on the TREC video corpus is shown in Table 1. The system performed extremely well for shot detection giving very high precision and recall.

	Ins. Rate	Del. Rate	Precision	Recall
Cuts	0.039	0.020	0.961	0.980
Gradual	0.589	0.284	0.626	0.715
All	0.223	0.106	0.831	0.893

Table 1: Shot boundary detection results for TREC video shot detection.

The results in Table 1 shows that the results for gradual changes could be improved. We found that in many of the cases, which were reported as errors, there was a detection of a boundary but the reported duration was too short. In such a case, the ISIS-based evaluation algorithm [ISI99] rejects the match, and considers it as both a deletion error and an insertion error. This is an undesired property of the evaluation criteria. If, for example, the system would not find a boundary at all, the evaluation would conider it as just a deletion, and rank the system better. In some other cases, a cut was reported as a short dissolve, with similar consequences.

Shot detection errors also resulted from the high noise level in the compressed MPEG video. For example, a periodic noisy pattern can be observed in Figure 3 at a period of 15 frames (one GOP) due to the color coding errors introduced by the MPEG encoding scheme. From our experience this noise level seemed somewhat high, but we have not quantified it.

4.3 Retrieval benchmark results

The results of the first retrieval experiment are shown in Table 2, which evaluates the average number of hits over the 46 "general search" queries. The interactive content-based retrieval (CBR) method is compared an automatic speech recognition (ASR) approach in which ASR was applied to the audio, and text indexing was used for answering the queries. The results show a signficant increase in retrieval quality using the interactive CBR approach.

Approach	Hits/query
Automatic speech recognition (ASR)	1.9
Interactive Content-based retrieval (CBR)	4.3

Table 2: Video retrieval results (avg. hits/query over 46 general searches).

Specific examples comparing retrieval performance for interactive CBR and ASR approaches are given in Table 3.

In some cases, such as topics VT66 and VT47, the ASR approach gave better retrieval results. In these topics, the relevant information was not easily captured by the visual scenes. However, for other topics, such as VT55, VT49, VT43, and VT42, the interactive CBR approach gave better performance than the ASR approach.

Topic#	Description	ASR	CBR
VT66	Clips about water project	9	3
VT47	Clips that deal with floods	8	1
VT55	Pictures of Hoover Dam	3	8
VT49	Lecture showing graphic	4	20
VT43	Shots showing grasslands	0	8
VT42	Shots of specific person	1	9

Table 3: Video retrieval results (hits/query) comparing interactive CBR and ASR methods for specific queries.

We also compared the interactive CBR approach to non-interactive (or automatic) CBR in which only a single iteration of searching was allowed. The results for two of the topics given in Table 4 show a significant increase in retrieval performance using the interactive CBR approach.

Topic	Description	Automatic	Interactive
#		CBR	CBR
VT54	Glen Canyon Dam	3	12
VT15	Shots of corn fields	1	5

Table 4: Video retrieval results (hits/query) comparing automatic and interactive CBR methods for specific queries.

5 Summary

In this paper, we described a system for automatic and interactive content-based retrieval that integrates features, models, and semantics. The system extracts feature descriptors from shots, which allows content-based retrieval, and classifies the shots using models for different events, scenes, and objects. The retrieval system allows the integration of content-based and model-based retrieval in an iterative search process. We developed also an approach based on speech indexing to provide a comparison with the content-based/model-based approach. We described the results of applying these methods to the TREC video retrieval benchmark.

References

[Cue] *IBM CueVideo Toolkit Version 2.1, http://www.almaden.ibm.com/cs/cuevideo/.* Download at http://www.ibm.com/alphaworks.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley Eastern, New York, 1973.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Proceedings of the Royal Statistical Society*, B(39):1–38, 1977.

[ISI99] In *European Workshop on Content Based Multimedia Indexing*, Toulouse, FR, October 1999. http://www-asim.lip6.fr/AIM/corpus/aim1/indexE.html.

[NCS+01] A. Natsev, Y.-C. Chang, J. R. Smith, C.-S. Li, and J. S. Vitter. Supporting incremental join queries on ranked inputs. In *Proc. Conf. on Very Large Databases (VLDB)*, Rome, Italy, September 2001.

[NKHR00] M. Naphade, I. Kozintsev, T. S. Huang, and K. Ramchandran. A factor graph framework for semantic indexing and retrieval in video. In *Proc. IEEE Workshop on Content-based Access to Image and Video Libraries (CBAIVL)*, Hilton Head, SC, June 12 2000.

[NLS+02] M. R. Naphade, C. Y. Lin, J. R. Smith, B. L. Tseng, and S. Basu. Learning to annotate video databases. In *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases 2002*, volume 4676, San Jose, CA, January 2002.

[Poo99] H. V. Poor. *An Introduction to Signal Detection and Estimation,*. Springer-Verlag, New York, 2 edition, 1999.

[PS01] D. Ponceleon and S. Srinivasan. Structure and content-based segmentation of speech transcripts. In *Proc. ACM Inter. Conf. Res. and Develop. in Inform. Retrieval (SIGIR)*, September 2001.

[RWSJ+95] 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.

[SBL+01] J. R. Smith, S. Basu, C.-Y. Lin, M. Naphade, and B. Tseng. Integrating features, models, and semantics for content-based retrieval. In *Proc. Multimedia Content-based Indexing and Retrieval (MMCBIR) workshop*, Rocquencourt, FR, September 2001.

[Smi01] J. R. Smith. MPEG-7 standard for multimedia databases. In *ACM Proc. Int. Conf. Manag. Data (SIGMOD)*, Santa Barbara, CA, May 2001. Tutorial.

[SS01] P. Salembier and J. R. Smith. MPEG-7 multimedia description schemes. *IEEE Trans. Circuits Syst. for Video Technol.*, August 2001.