IBM Research TRECVID-2003 Video Retrieval System

Arnon Amir[3], Marco Berg[3], Shih-Fu Chang[4], Winston Hsu[4], Giridharan Iyengar[2], Ching-Yung Lin[1], Milind Naphade[1], Apostol (Paul) Natsev[1], Chalapathy Neti[2], Harriet Nock[2], John R. Smith[1], Belle Tseng[1], Yi Wu[1], Donqing Zhang[1]

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

[2] IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

[3] IBM Almaden Research Center
650 Harry Rd
San Jose, CA 95120

[4] Columbia University
E.E. Department
New York, NY 10027

Abstract

In this paper we describe our participation in the NIST TRECVID-2003 evaluation. We participated in four tasks of the benchmark including shot boundary detection, high-level feature detection, story segmentation, and search. We describe the different runs we submitted for each track and discuss our performance.

1. Introduction

Content-based retrieval of video presents significant challenges in terms of development of effective techniques for analysis, indexing and searching of video databases. TRECVID is greatly facilitating the advancement of technologies for content-based retrieval of video by providing a standard dataset and evaluation forum for evaluating emerging and novel techniques and systems. The IBM team participated in TRECVID for the third time since its inception in 2001. This year, Columbia University joined our team to work on the story segmentation and high-level feature detection tasks. The goal of our participation in 2003 was to participate in all four of the TRECVID tasks – shot boundary detection, high-level feature detection, story segmentation, and search (manual and interactive) – and to explore large variation of techniques for each task. As a result, we developed a wide range approaches and systems, and we submitted the maximum number of runs for each task. We also participated in organizing a new TRECVID-wide effort this year to jointly develop an annotated video dataset to be used for system development. This effort was very successful in creating a richly annotated dataset of 63 hours of video, which was used by participants to train classifiers for the high-level feature detection and search tasks. The benefits of this training dataset are numerous, including that fact that better comparisons could be drawn across systems based on inherent techniques rather than dependencies on particular training practice.

2. Annotation Forum

We developed a *VideoAnnEx* MPEG-7 annotation tool to allow TRECVID participants to semi-automatically annotate video content with semantic descriptions [5][6][7]. The tool explores a number of interesting capabilities including automatic shot detection, key-frame selection, automatic label propagation to similar shots, and importing, editing, and customizing of ontology and controlled term lists.

Given the lexicon and video shot boundaries, visual annotations can be assigned to each shot by a combination of label prediction and human interaction. Labels can be associated to a shot or a region on the keyframe. Regions can be manually selected from the keyframe or injected from the segmentation module. Annotation of a video is executed shot by shot without permuting their time order, which we consider an important factor for human annotators because of the time-dependent semantic meanings in videos. Label prediction utilizes clustering on the keyframes of video shots in the video corpus or within a video. By the time a shot is being annotated, the system predicts its labels by propagating the labels from the last shot in time within the same cluster. Annotator can accept these predicted labels or select new labels from the hierarchical controlled-term lists. All the annotation results and descriptions of ontology are stored as MPEG-7 XML files.

VideoAnnEx v2.0 allows collaborative annotation among multiple users through the Internet. Users of the collaborative VideoAnnEx are assigned user IDs and passwords to access a central server, called the VideoAnnEx CA (collaborative annotation) Server. The VideoAnnEx CA Server centrally stores the MPEG-7 data files, manages the collaboration controls, and coordinates the annotation sessions. For collaborative annotation, there are three categories of user access to the VideoAnnEx CA Server, and they are: (1) project manager, (2) group administrator, and (3) general user. The project manager sets up the project on the VideoAnnEx CA Server, creates the different groups' IDs and allocates video responsibilities to groups. The group administrator coordinates the annotations of

the assigned videos and distributes the annotation tasks among the individual general users. The general users are the end users who actually perform the annotation task on the VideoAnnEx v2.0 Client.

Using VideoAnnEx Collaborative Annotation System as the infrastructure, we initiated and organized a Video Collaborative Annotation Forum of TRECVID 2003. The objective of this forum was to establish ground-truth labels on large video datasets as common assets to research society. The first phase of the forum was to annotate labels on the NIST TREC Video Retrieval Evaluation 2003 (TRECVID) development video data set. This development video data is part of the TRECVID 2003 video data set which includes:

- 120 hours (241 30-minute programs) of ABC World News Tonight and CNN Headline News recorded by the Linguistic Data Consortium from late January through June 1998 and
- 13 hours of C-SPAN programming (~ 30 mostly 10- or 20-minute programs) about two thirds 2001, others from 1999, one or two from 1998 and 2000. The C-SPAN programming includes various government committee meetings, discussions of public affairs, some lectures, news conferences, forums of various sorts, public hearings, etc.

The total TRECVID 2003 video set is about 104.5 GB of MPEG-1 videos, that includes the development set (51.6 GB, 62.2 hours including 3390 minutes from ABC & CNN, 340 minutes from C-SPAN) and the test set (52.9 GB, 64.3 hours including 3510 minutes from ABC & CNN, 350 minutes from C-SPAN). TRECVID 2003 participants have the option to join the Video Collaboration Annotation Forum, which establishes the common annotation set that all forum participants agree to contribute annotations. Based on these common development set and common annotation set, forum participants can develop Type 1 (as specified by NIST) feature/concept extraction system, search system or donation of extracted features/concepts. This set of common annotation will be available to the public after the TRECVID 2003 workshop in November 2003. From April to July 2003, 111 researchers from 23 institutes worked together to associate 197K of ground-truth labels (433K after hierarchy propagation) at 62.2 hours of videos. 1038 different kinds of labels were annotated on 46K manually aligned shots. Details and User Study of the video collaborative annotation forum can be seen at [5]

3. Shot Boundary Detection

The IBM shot boundary determination (SBD) system at TRECVID 03 is an improved version of the CueVideo SBD algorithm used in TRECVID 02. This algorithm is based on a Finite States Machine (FSM), processing one frame at a time in a single pass over the video. It uses RGB color histograms, localized edge intensity histograms and thumbnails comparisons to evaluate the difference between pairs of frames at 1,3,5,7,9 and 13 frames apart. Adaptive thresholds are computed using rank filtering on pairs differences statistics in a window of 61 frames around the processed frame. A different threshold is computed for pairs of frames at different time-differences.

3.1. Approach

The manually-crafted FSM algorithm allows for local improvements to address specific types of errors. First, a baseline system is set up. A training set of ten video segments was constructed from the Search training set. Each segment is 5 minutes long, and its location within the video is pseudo-randomly selected from a uniform distribution. Ground truth for the training set was manually constructed. The baseline system was applied and the errors were visited to see the most common causes of error. Those were addressed one at a time to improve the FSM. The main algorithmic improvements in 2003 are:

- A flash photography detector. Eliminates false detections on rapid flashlights. This was counted as the number one cause of cuts insertion errors in the training set.
- Better handling of fades, including some tuning to improve detection, linking fade out-in, and handling of abrupt fades, which start as a cut to black, followed by a gradual fade-in.
- Improved detection of graduals' boundaries. We analyzed the amount of frame errors at the beginning and end of gradual boundaries and tuned each one separately.
- Detection & handling of partial-frame MPEG errors. Apparently there were several videos with bad macro-blocks. Detecting those and eliminating false detection of a cut reduces insertion errors.

The flash detection is based on finding an abrupt change in the frame illumination compared to frames before and after it. Figure 1 shows a few examples – the first two are correct detections while the last one (ranked 58) shows a failure mode (a single bright frame between two shots). The detection of a flash suppresses the detection of a cut at that place. MPEG errors are handled differently by different MPEG decoders. Previously, our algorithm handled

only full-frame errors. This year we added detection of partial frame errors and elimination of false detection at the boundaries of sub-frame errors. Such errors may occur in multiple frames.

3.2. Results

The overall SBD results are summarized in Table 1. The best IBM run, labeled n127, was better than any non-IBM run in all four evaluation measures – All Transitions, Cuts, Graduals and Gradual Frame Accuracy. Overall, the 7 top runs in each measure are of IBM. The improvement from the baseline system is noticeable mainly in the gradual changes – both detection and accuracy.

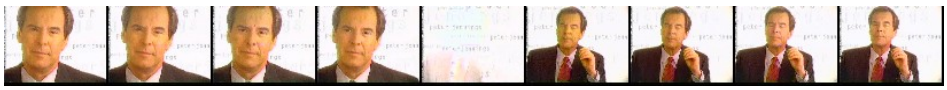
Rank	Frame#	Conf.	Consecutive frames around the detected flashlight
1	1394	1.00	
2, 3	20067-20068	1.00	
58	49679	0.39	

Figure 1. Example: abrupt flashlights detected by the algorithm in 19980104_ABC.mpg. A total of 133 flash events are detected in this video, of which 122 are correct.

Table 1 - SBD results at TRECVID 03. Top 30 (out of 76) runs at each of the four evaluation measures are listed in decreasing F-number order. The ten IBM runs are highlighted– IBM best run at TRECVID 03, 02 and 01 (n127-yellow, n047-beige ,nalm1-purple), baseline run (n110, cyan) , and the six other IBM runs (blue). All other runs are marked with NIST marker conventions.

	All Transitions				Cuts				Graduals				Graduals frame accuracy			
	Sys	Rcl	Prc	F#	Sys	Rcl	Prc	F#	Sys	Rcl	Prc	F#	Sys	Rcl	Prc	F#
1	n127	0.892	**0.925**	0.908	n128	0.935	0.951	0.943	n119	0.792	0.865	0.827	n123	0.823	0.875	0.848
2	n119	0.894	0.921	0.907	n127	0.937	0.947	0.942	n127	0.784	0.865	0.823	n128	0.794	0.893	0.841
3	n120	0.899	0.913	0.906	n123	0.936	0.946	0.941	n120	0.802	0.840	0.821	n120	0.793	0.894	0.840
4	n122	0.899	0.913	0.906	n120	0.938	0.942	0.940	n122	0.802	0.840	0.821	n122	0.793	0.894	0.840
5	n128	0.895	0.917	0.906	n122	0.938	0.942	0.940	n128	0.798	0.833	0.815	n127	0.771	0.913	0.836
6	n126	0.887	0.919	0.903	n119	0.936	0.942	0.939	n126	0.776	0.856	0.814	n119	0.784	0.894	0.835
7	n123	0.899	0.906	0.902	n126	0.932	0.943	0.937	n123	0.809	0.809	0.809	n126	0.723	0.913	0.807
8	nalm1	0.916	0.868	0.891	z	0.893	**0.976**	**0.933**	nalm1	**0.840**	0.761	0.799	a	**0.835**	0.759	0.795
9	n110	0.894	0.881	0.887	nalm1	0.947	0.916	0.931	n110	0.784	0.804	0.794	c	0.742	0.844	0.790
10	n047	0.896	0.875	0.885	n110	0.939	0.911	0.925	n047	0.778	0.808	0.793	c	0.748	0.835	0.789
11	c	0.882	0.881	0.881	n047	0.944	0.900	0.921	c	0.737	0.849	0.789	c	0.761	0.812	0.786
12	c	0.863	0.898	0.880	c	0.942	0.891	0.916	c	0.713	0.882	0.789	n110	0.681	0.905	0.777
13	c	0.892	0.858	0.875	c	0.960	0.872	0.914	c	0.693	0.913	0.788	n047	0.681	0.899	0.775
14	c	0.829	0.921	0.873	c	0.905	0.917	0.911	c	0.742	0.827	0.782	c	0.762	0.788	0.775
15	c	0.904	0.832	0.867	s	0.958	0.868	0.911	c	0.672	0.924	0.778	a	0.812	0.737	0.773
16	o	0.845	0.868	0.856	r	0.961	0.855	0.905	c	0.755	0.784	0.769	a	0.789	0.754	0.771
17	c	0.909	0.809	0.856	c	0.966	0.851	0.905	c	0.645	0.934	0.763	a	0.784	0.750	0.767
18	o	0.839	0.871	0.855	o	0.910	0.892	0.901	c	0.762	0.745	0.753	c	0.755	0.770	0.762
19	o	0.855	0.854	0.854	o	0.911	0.889	0.900	o	0.698	0.805	0.748	a	0.771	0.735	0.753
20	o	0.855	0.839	0.847	o	0.905	0.890	0.897	o	0.717	0.762	0.739	c	0.756	0.749	0.752
21	o	0.863	0.826	0.844	v	0.940	0.855	0.895	c	0.783	0.697	0.738	a	0.798	0.704	0.748
22	o	0.800	0.890	0.843	o	0.911	0.880	0.895	o	0.734	0.733	0.733	v	0.612	0.952	0.745
23	c	0.914	0.777	0.840	z	0.835	0.963	0.894	o	0.664	0.808	0.729	c	0.749	0.733	0.741
24	o	0.807	0.874	0.839	c	0.972	0.827	0.894	o	0.643	0.834	0.726	v	0.598	0.950	0.734
25	o	0.788	0.897	0.839	o	0.905	0.882	0.893	o	0.749	0.699	0.723	c	0.750	0.718	0.734
26	o	0.866	0.801	0.832	a	0.918	0.869	0.893	o	0.608	0.885	0.721	v	0.596	0.947	0.732
27	a	0.794	0.868	0.829	a	0.918	0.868	0.892	c	0.787	0.645	0.709	nalm1	0.597	0.928	0.727
28	a	0.792	0.870	0.829	o	0.910	0.868	0.889	o	0.657	0.768	0.708	v	0.586	0.944	0.723
29	o	0.808	0.847	0.827	a	0.924	0.855	0.888	o	0.760	0.654	0.703	v	0.583	0.944	0.721
30	c	0.916	0.753	0.827	a	0.924	0.855	0.888	o	0.564	0.907	0.696	v	0.583	0.941	0.720

4. High-Level Feature Detection

The IBM TREC 2003 Concept Detection Framework consists of several sequential processing modules or silos. We will refer to the sequencing of these silos as the Concept Detection pipeline. Typically, each level of processing uses a training set and a validation set.

4.1. Training Design

Depending on the nature of these modules the data sets on which training and validation is done vary. To support this we divided the development data corpus of 60 hours of MPEG-1 video into 4 distinct non-overlapping parts. The quantum of partitioning is at the video clip level. The first set hereby referred to as the training set consists of 60 % of the development data. The next set referred to as Validation Set 1 consists of 10 % of the development data. The third set referred to as Validation Set 2 consists of 10 % of the development data. Finally the fourth set referred to as the Validation Set 3 consists of 20 % of the development data. The division is accomplished by sequentially assigning proportional number of video clips to each set as we go forward in temporal order. This ensures that the time-span of the training set finds representation in all the four sets. We will describe the need for this division and the utilization of the data sets next.

4.2. Concept Detection Pipeline

The IBM TRECVID 2003 Concept Detection Pipeline is shown in Figure 2. We begin with the annotated development corpus partitioned into 4 sets. The first silo extracts low-level multimedia features from the data. Features are extracted from key-frames, regions within key-frames obtained either through bounding box mark-up or through segmentation. Visual Features include color correlograms (CC), color histograms (CH), edge histogram (EH), moment invariants for shape (MI), motion vectors extracted from multiple frames within a shot (MV), wavelet texture features (WT), Tamura texture features (TAM) and cooccurrence textures (CT). Aural features include MFCC coefficients.

The second silo contains unimodal feature representations learnt using the training data and tuned for optimization over Validation Set 1. The modules in this silo include Speech-based models, (SD/A), and support vector machine (SVM) based visual models (V1, V2) developed for two different feature subsets. This modeling results in multiple representations for the 17 TREC benchmark concepts as well as 47 secondary concepts that belong to the common annotation lexicon and are thought to be related to the 17 benchmark concepts, as well as relevant to possible search topics in this domain. For details on the SVM-based modeling we refer the readers to [24]. Details of speech-based modeling can be found in [12]. In the case of visual SVM models, the training set is used to build models for a variety of parametric configurations for each concept. The validation set is then used to choose the one parametric configuration for each concept that leads to the highest achievable average precision on the validation set. One set of SVMs are trained using early feature fusion of color, texture, edge and shape features. The other set of SVMs are trained using a high-dimensional color histogram, Tamura texture, and other features. At this silo, several models for each concept are thus created, where each model is based on features extracted either from the visual or the audio modality.

The job of the processing modules in the third silo is to then take the soft decisions obtained by using the models in the second silo and fuse all such decisions for each concept. The modules in this silo fuse the detection for all models belonging to the same concept across different modalities, and feature combinations to generate one detection list per concept. However there are several different ways in which this fusion can be achieved. The silo thus contains three different modules which consume the detection of the second silo and each creates one decision list per concept. All modules in this silo use the Validation Set 1 for training their models and use Validation Set 2 for selecting optimal parametric configuration. The fusion modules include ensemble fusion (EF), validity weighting (VW), and a multi-layered perceptron (MLP) applied for fusing models for the 17 benchmark concepts. For the 47 secondary concepts, only ensemble averaging is applied (This is a result of lack of processing time and not of any systematic belief that the other two fusion methods would not work for the 47 secondary concepts.). Ensemble fusion works with different normalization schemes and uses averaging to combine multiple decision lists. Validity Weighting is a weighted averaging scheme where the weight of any model is proportional to its performance. Here validity weighting uses the average precision of the models from the second silo on Validation Set 1 to determine the weighting. This scheme thus favors models that perform better while penalizing models which perform worse. The Neural network approach treats the fusion problem as a classification problem using the validation set 1 as its training set and validation set 2 as the set used for selecting optimal parametric configurations.

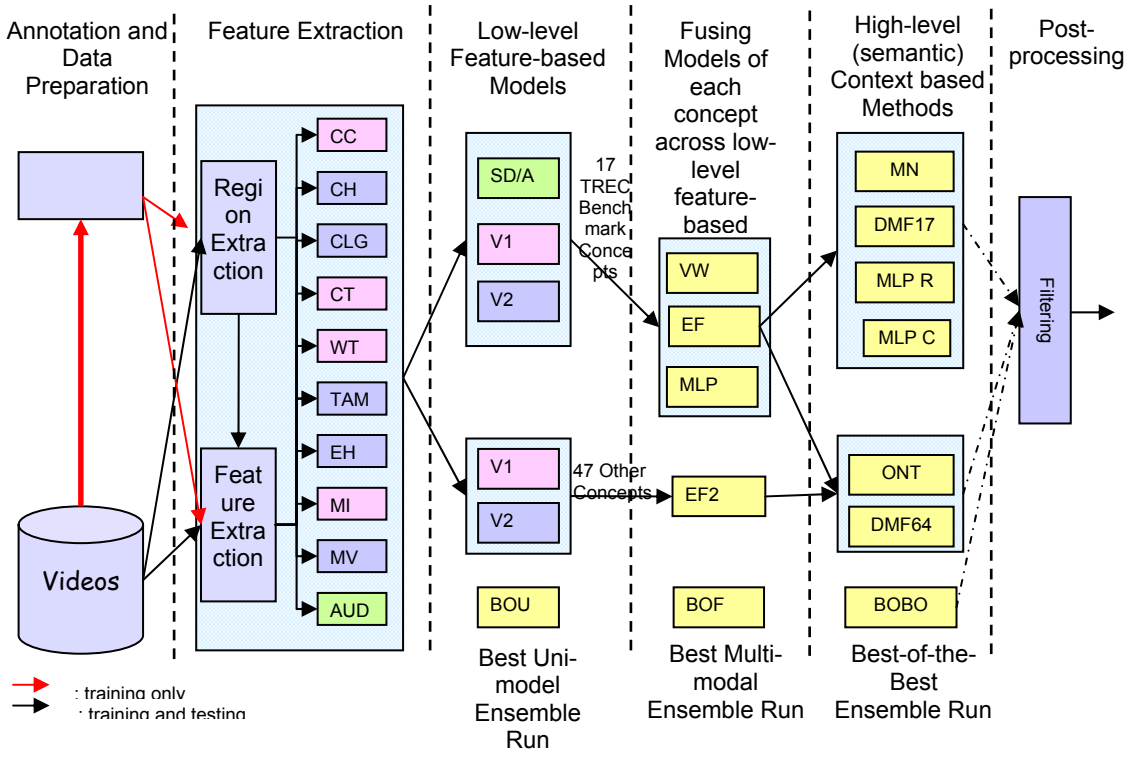


Figure 2. The IBM TRECVID 2003 Concept Detection Pipeline. Data flows from left to right through various silos. At each silo, multiple parallel paths of processing are available taking the data through the steps of low-level media feature extraction in the first silo, training of aural, visual and speech-based models for 64 concepts in the second silo, fusion of multiple models for identical concepts in the third silo, context enforcement and model-vector space manipulation for improving performance in the fourth silo and filtering for improving precision in the fifth silo.

The fourth silo in our processing pipeline is the fusion of models across semantic concepts. This is an attempt to use the dependencies between concepts to improve the detection. The input of each module in this silo is the detection results generated by the previous fusion silo. All the modules in this silo work in the model vector space, created by concatenating the detection confidences of the various concepts for which models have been built. In particular we use 5 different model-vector based fusion techniques. The first technique is the multinet (MN) approach of Naphade et al [NSC03], which uses loopy probability propagation in the sum product algorithm in a factor graphical setting to account for the correlation between different concepts and cyclic nature of the inter-conceptual dependencies. We also try the classical classification approach, where model vectors are treated like any other feature vectors and models are then built for each concept again in this space using supervised training. Two variants of this approach termed as discriminative model fusion (DMF17, DMF64) are developed using the 17 benchmark concepts (DMF17) and the 64 concepts modeled (DMF64). Support vector machines are used as the classifier for each concept in both these variants. A third variant (MLP C) uses neural networks instead of support vector machines. Validation Set 2 is used for training and validation set 3 is used for optimal parameter selection. Another approach is to approximate the vector space of decision outputs using a regression like approach over the model vectors. A multi-layered perceptron (MLP R) is applied for this regression approach. This also uses validation set 2 for training and validation set 3 for optimal parameter selection. Finally we also try to use the knowledge already present in the manually defined hierarchy in the annotation lexicon (ONT). Using this hierarchy we modify decisions for leaf nodes (which correspond to rarer concepts) based on the detection of their parents (which are root nodes corresponding to more frequent concepts for which better detection results exist).

In each silo we also evaluate the best performer for each concept from amongst all the modules in the silo and save this as the "Best-of (BO) performer". In the unimodal processing silo this is labeled best of unimodal processing (BOU). In the single concept fusion silo, this is termed as best of fusion (BOF) processing. In the model-vector processing silo this is termed as best of the best (BOBO) because at this stage we are in a position to measure the best performing module across all silos. After the data passes through all these processing modules in the four silos, we finally filter it to improve precision. Typical filtering tasks, involve pushing down the C-SPAN videos for

sports event detection etc. This is done using matched filtering for commercials, and for other content characteristics such as the nature of the video clip (C-SPAN, CNN, ABC) etc.

4.3. Feature extraction

We developed numerous feature extractors based on descriptors for color, texture, edges, and motion. Many of these descriptors were based on the ones defined in the MPEG-7 standard. The system extracts two sets of visual features for each video shot. These two sets are applied by two different modeling procedures that are described in Section 3.5. The first set includes: (1) color histogram (YCbCr, 8x3x3 dimensions), (2) Auto-correlograms (YCbCr, 8x3x3 dims), (3) Edge orientation histogram (32 dim), (4) Dudani's Moment Invariants (6 dims), (5) Normalized width and height of bounding box (2 dims), (6) Co-occurrence texture (48 dims). These visual features are all extracted from the keyframe.

The other set of visual features include: (1) Color histogram (RGB, 8x8x8 dims), (2) Color moments (7 dims), (3) Coarseness (1 dim), (4) Contrast (1 dim), (5) Directionality (1 dim), and (6) Motion vector histogram (6 dim). The first five features are extracted from the keyframe. The motion features are extracted from every *I* and *P* frames of the shot using the motion estimation method described in the region segmentation module. Depending on the characteristics, some of the concepts are considered as global, such as outdoors, indoors, factory setting, and office setting, while some of them are regional, such as sky, mountain, greenery, and car. Therefore, we extract these features on both the frame level and the region level.

4.4. Uni-modal detectors

We developed several uni-modal detectors that perform concept detection using speech and visual information.

4.4.1. Speech

Automatic Speech Recognition (ASR) transcript of the training data is analyzed in and near the occurrence of the target concepts to extract a list of words. This list is further refined manually to select a set of query terms for the particular concept. Test data transcripts are indexed similar to the Spoken Document Retrieval (SDR) systems using OKAPI. For concept detection, the above chosen query words are used in the SDR system to retrieve shots which are then annotated with the appropriate concept.

4.4.2. Visual

Two sets of unimodal visual models were trained using support vector machine (SVM). The SVM-light software[24] was used for the experiments reported. The first process uses the Concept Training (CT) set to training classifiers and uses Concept Validate (CV) set to select the best parameters for each individual SVM classifier. This process generates a SVM classifier for each type of visual feature, e.g., one classifier based on color histogram, one based on moments, and so forth. It also generates SVM classifiers based on heuristic combinations of features. The second modeling procedure is similar to the first procedure, except that a different set of visual features is used and the CV set is not used for parameter selection. The visual concept modeling modules of the second silo is performed using support vector machine classifiers. For each concept a binary hypothesis approach is taken to concept detection. Presence or absence of the concept is determined from the training set annotation. For each concept, we build a number of SVM models for different parametric configurations. In all 27 parametric configurations are modeled at a minimum for each concept. Two sets of SVM models are built for disjoint feature vectors. In one set the feature vector is a concatenated early feature fusion based vector consisting of color correlogram, co-occurrence texture features, edge histogram and moment invariants. Depending on the nature of the concept these features are extracted either globally or at regional level. For the training set the regions are obtained from the bounding boxes marked up by the annotators. For the three validation sets as well as the search test set, the regions are obtained by segmentation and automatic detection of up to 5 bounding boxes for each key-frame.

4.5. Fusion models

The fusion models combine the output of multiple unimodal models. We explored fusion techniques based on ensemble fusion and validity weighting.

4.5.1. Ensemble fusion

The normalized ensemble fusion process consists of three major steps. The first step is normalization of resulting confidence scores from each classifier. The second step is the aggregation of the normalized confidence scores. The third step is the optimization over multiple score normalization and fusion functions to attain optimal performance. Each classifier generates an associated confidence score for the data in the validation set. These confidence scores are normalized to a range of [0, 1]. The normalization schemes include: (1) rank normalization, (2) range

normalization, and (3) Gaussian normalization. After score normalization, the combiner function selects a permuted subset of different classifiers and operates on their normalized scores. We essentially identify the high-performing and complementary subsets of classifiers. Following, different functions to combine the normalized scores from each classifier are considered. The combiner functions we tried include: (1) minimum, (2) maximum, (3) average and (4) product. Subsequently, an optimal selection of the best performing normalized ensemble fusion is obtained by evaluating the average precision (AP) measure against the CV ground truth. We will then use the fusion parameters, which optimize the outcome of CV set, on the next stage of test set (i.e. CF1 set). This approach has been applied on the uni-modality/uni-model fusion, multi-modality/uni-model fusion, and multi-modality/multi-model fusion.

4.5.2. Validity weighting

While model vector-based classification is a relatively simple yet powerful approach, techniques are needed to ensure good performance when a potentially large set of detectors are used and the quality or validity of the detectors varies. We studied the case in which the validity weightings are based on the number of training examples used to build each underlying detector.

4.6. Context models

While each semantic concept can be modeled directly from the positive and negative training samples. However such direct modeling overlooks the presence of contextual constraints within concepts that occur together. Intuitively it is obvious that the presence of certain semantic concepts suggests a high possibility of detecting certain other multijects. Similarly some concepts are less likely to occur in the presence of others. The detection of sky and greenery boosts the chances of detecting a Landscape, and reduces the chances of detecting Indoors. It might also be possible to detect some concepts and infer more complex concepts based on their relation with the detected ones. Detection of human speech in the audio stream and a face in the video stream may lead to the inference of human talking.

4.6.1. Multinets

To integrate all the concepts and model their interaction, we proposed the network of multijects or multinets [18]. We demonstrated the application of the multinet based context enforcement for TREC 2002 earlier [22]. We applied the multinet model to TRECVID 2003 Concept Detection. The actual implementation uses a factor graphical framework which is suitable for loopy probability propagation and can account for undirected (non-causal) relationships between different semantic concepts. We also apply temporal modeling of context in addition to the conceptual constraints. This approach resulted in the highest MAP performance across all the TRECVID 2003 concept detection runs. To summarize the processing, the multinet accepts as input the model vector or the individual concept detection for all the concepts. It then learns from the training set the correlation relationships between all the concepts as well as between pairs of concepts. The pair-wise relationship modeling helps improve efficiency of the multinet during the inference phase. The concept detection is then modified based on these correlation constraints through loopy or iterative propagation of the probability of concept detection. The propagation is loopy because the graph has several cycles indicating the complex relationship between concepts. At the end of the propagation iterations the detection for each concept is thus modified based on its relationship with other concepts and the detection confidence of these other concepts.

4.6.2. Regression-based context models

One of the approaches we considered for semantic context exploitation was that of building simple regression models of all target concepts in terms of their relationships with the other concepts. This is especially suitable for rare concepts which are difficult to model robustly directly from the low-level visual or audio features. If these concepts are highly correlated to other more frequent concepts—which can be modeled robustly—then we could leverage those correlations and build simple models of the rare concepts in terms of the more frequent ones. Given an existing basis of robust semantic models, a very simple and intuitive approach is to build new models as weighted combinations of the basis ones. For example, the weights can be set proportionally to the correlation coefficients of the modeled concept with respect to each of the basis concepts. Given a weighted linear formulation, the weights can in fact be recovered optimally (in the mean squared error sense) from a training set using the least squares fit method. This formulation poses certain independence constraints on the semantic basis which may not be fulfilled in practice, however. We therefore considered general regression models (both linear and non-linear) and used multi-layer perceptron (MLP) neural networks to learn the regression models. We used a single hidden layer and optimized the number of hidden nodes and their activation functions (linear, sigmoid, or tanh), selecting the optimal parameter configuration on an independent validation set. We built MLP-based regression models for all target

concepts from the NIST feature detection task except for Zoom In, resulting in a run with the highest overall Mean Average Precision on a separate validation set that was used internally for evaluation of submitted IBM runs. The performance of the regression models did not generalize as well to the NIST Search set, however. This can be partially explained by the inconsistent performance (between the development and search sets) of one of the input basis models, which was highly correlated to several of the modeled concepts.

4.6.3. Discriminative model fusion

The Discriminative Model Fusion approach starts with a set of "basis" concept models and further refines those models or builds new concept models. Given a set of trained concept models, they are run on a separate training data partition to generate scores for these concepts. These scores are grouped together into a vector (called the *model vector*) and are normalized to have zero-mean and unit variance [14]. On this training data partition, new concept models are discriminatively trained using the model vectors as input features and a support vector machine (SVM) learner to learn a binary classification. We note here that the binary classifiers can learn concept models for existing concepts (we refer to these as the "in-basis" set) or can learn previously unseen concept classes (the "out-of-basis" set). For TRECVID 2003, the models learned were based on the model vectors derived from the Ensemble Fusion model set and were chosen to be in-basis. Two types of DMF concept models were trained – The first set comprised model vectors derived from 17 primary concepts (DMF17); The second set comprised model vectors from these 17 primary concepts and 47 secondary concepts (DMF64). Our experiments with the TRECVID 2003 corpus indicated that in terms of MAP, the two DMF runs did not improve upon the Ensemble Fusion models. However, in terms of individual models, DMF based models had the best P@100 for 5 out of the 17 benchmarked concepts and best AP numbers for 4 of the 17 concepts. In contrast, for the 10 TRECVID 2002 concepts, we noted a 12% gain in terms of MAP over the baseline models.

4.6.4. Ontology

NIST TREC-2003 Video Retrieval Benchmark defines 133 video concepts, organized as an ontology hierarchy. The task of video concept detection is to classify each data (shot) into corresponding concepts. One shot can have multiple concepts, thus it is a multi-classification problem requiring inputs to be mapped into one or several possible categories. Ontology-based multi-classification learning consists of two steps. At the first step, each single concept model is constructed independently using Support Vector Machines (SVMs). At the second step, ontology-based concept learning improves the accuracy of individual concept by considering the possible influence relations between concepts based on predefined ontology hierarchy.

There are two kinds of influences defined in ontology-based learning. One is *boosting factor* and the other is *confusion factor*. *Boosting factor* is the top-to-down influence from concepts located at upper level of ontology hierarchy to their semantically related concepts located at lower level. From our experiments on single concept models, we have observed that for concepts located at the lower levels of the ontology hierarchy, their corresponding classifiers are likely to get lower accuracy compared with those classifiers on the upper level with more positive training data. The main idea of *boosting factor* is to boost the precision of concepts by taking influences from more reliable ancestors. *Confusion factor* is the influence between concepts that cannot be co-existent. The main idea of *confusion factor* is to decrease the probability of misclassifying data s into class C_i however s truly belongs to C_k, and C_k and C_i cannot coexist in semantics. For example, if data s has semantic concept of "Studio setting", it cannot belong to concept "Non-Studio setting". Ontology hierarchy decides the influence paths of *boosting factor* and *confusion factor* for each concept, and the influence values of *boosting factor* and *confusion factor* are decided based on data correlation.

The advantage of ontology learning is that its influence path is based on ontology hierarchy, which has real semantic meanings. Besides semantics, ontology learning also considers the data correlation to decide the exact influence assigned to each path, which makes the influence more flexible according to data distribution.

4.7. Post-Filtering

We developed a news detector that is used to filter the concepts that should not appear in the news domain such as non-studio setting, or should be in the news domain such as Albright, Weather, Sports, etc. The basic structure of the news detector is shown on Figure 3. We used color and edge features for template matching. Only the regions that correspond to the location of templates are tested, and the result S is a binary decision on the test frames.

$$S = \delta(S_C > \tau'_C) \,\&\, \delta(S_E > \tau'_E), \text{ and}$$

$$S_C = \frac{1}{N}\sum_n \delta(d(P_C, P_{MC}) > \tau_C) \text{ and } S_E = \frac{1}{N}\sum_n \delta(d(P_E, P_{ME}) > \tau_E)$$

where C represents the color features and E represents the edge features. Four thresholds $\tau_C, \tau_E, \tau'_C, \tau'_E$, were used. $\delta()$ is the binary decision function, and d() represents the Euclidean distance of the test regions in the feature space. N is the number of pixels in that region. After binary decisions were made to the individual shots in a video, two consecutive temporal median filters were used to eliminate randomly false classified shots. Median filters The window size of both media filters is five shots.

We got a reasonable detection rate based on one template for CNN news and five templates for ABC news, (including two for ABC logo detection and three for closed caption areas at inter-story shots) These templates were randomly chosen in the training set. All templates used the same threshold, which is selected based on testing of two other videos in the testing set. We evaluate these news detectors on the CV set. The Misclassification *(Miss + False Alarm)* shots are 8 out of 1790 shots (accuracy = 99.6%) in the CNN videos and 60 out of 2111 shots (accuracy=97.2%) in the ABC videos. This news detector is also used to determine the type of news segments for the story segmentation task.

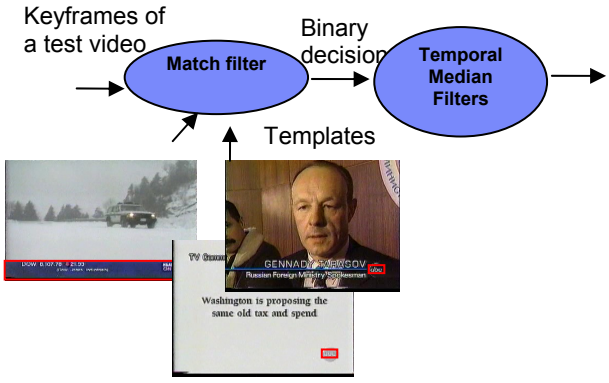


Figure 3. Post-Filtering News Detector

4.8. Results

Ten runs were submitted by The IBM TRECVID Concept Detection Pipeline. Nine of these runs were the top 9 performers in terms of mean average precision (MAP). The tenth run was ranked 11th in terms of MAP performance. Columns in Table 2 are organized from left to right following the order of data flow from left to right in the concept detection pipeline in Figure 2. The first column BOU illustrates the best result possible at the second processing silo. The next two columns (columns 2 and 3) EF and BOF, illustrate results at the end of the third processing silo. The next 7 columns illustrate results at the end of the fourth model-vector processing silo. Of these the three classification based modules are clubbed together (DMF17, DMF64 and MLP C). Note that the BOBO is the best of all modules at all processing stages. The fact that BOBO does not have the highest MAP indicates that the module that was selected to be the best based on the validation set 3 performance, did not generalize to perform as well on the search test set. Since amount of processing increases from left to right among the three sets delineated by the black, olive green and dark red font colors, and we have highlighted the best performance for each concept achieved using minimum processing

Table 2 - Comparing performance of the submitted IBM runs for all 17 benchmark concepts.

	BOU	EF	BOF	DMF17	DMF64	MLPC	MLPR	ONT	MN	BOBO
Outdoors	0.138	0.153	0.172	0.198	0.185	0.164	0.195	0.153	0.227	0.227
News Subject Face	0.161	0.176	0.176	0.179	0.108	0.115	0.182	0.176	0.096	0.108
People	0.217	0.244	0.245	0.221	0.109	0.246	0.254	0.244	0.24	0.24
Building	0.09	0.12	0.151	0.109	0.118	0.052	0.117	0.128	0.118	0.128
Road	0.115	0.1	0.181	0.175	0.144	0.095	0.171	0.118	0.183	0.181
Vegetation	0.271	0.392	0.393	0.394	0.347	0.39	0.379	0.393	0.392	0.393
Animal	0.02	0.213	0.213	0.173	0.102	0.193	0.203	0.213	0.213	0.02
Female Speech	0.075	0.143	0.143	0.043	0.207	0.022	0.059	0.143	0.134	0.134
Car Truck or Bus	0.194	0.276	0.276	0.243	0.159	0.119	0.213	0.27	0.285	0.159
Aircraft	0.237	0.403	0.296	0.428	0.394	0.384	0.409	0.412	0.404	0.404
NewsSubject Monologue	0.043	0.043	0.043	0.007	0.001	0.005	0.005	0.006	0.043	0.001
NonStudio Setting	0.073	0.129	0.087	0.1	0.067	0.066	0.117	0.126	0.129	0.086
Sports Event	0.467	0.708	0.642	0.642	0.618	0.682	0.706	0.708	0.699	0.642
Weather	0.548	0.849	0.848	0.843	0.814	0.839	0.833	0.849	0.849	0.856
Zoom In	0.118	0.118	0.118	0.118	0.118	0.118	0.118	0.118	0.118	0.118
Physical Violence	0.076	0.055	0.085	0.016	0.086	0.084	0.051	0.055	0.03	0.03
Madeline Albright	0.316	0.314	0.301	0.313	0.042	0.236	0.286	0.343	0.314	0.301
MAP	0.186	0.261	0.257	0.247	0.213	0.224	0.253	0.262	0.263	0.237

Table 2 reveals several interesting observations. Processing beyond single classifier per concept improves performance. If we divide TREC Benchmark concepts into 3 types based on frequency of occurrence Performance of Highly Frequent (>80/100) concepts is further enhanced by Multinet (e.g. Outdoors, Nature Vegetation, People etc.). Performance of Moderately Frequent concepts (>50 & < 80) is usually improved by discriminant reclassification techniques such as SVMs (DMF17/64) or NN (MLP_BOR, MLP_EFC). Performance of very rare concepts needs to be boosted through better feature extraction and processing in the initial stages.

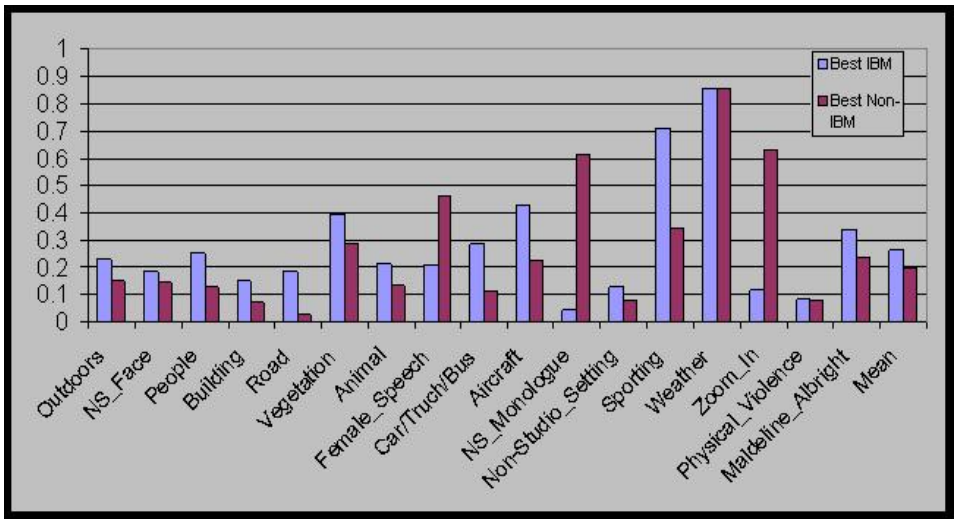


Figure 4. Comparing IBM Concept Detection with other submitted runs at NIST. The IBM Concept Detection Runs result in highest average precision in 14 of the 17 concepts, and also in the highest mean average precision corresponding to the multinet based run.

Based on Fusion Validation Set 2 evaluation, visual models outperform audio/ASR models for 9 concepts while the reverse is true for 6 concepts. Semantic-feature based techniques improve MAP by 20 % over visual-models alone. Fusion of multiple modalities (audio, visual) improves MAP by 20 % over best unimodal (visual) run (using Fusion Validation Set II for comparison). Figure 4 compares the best IBM run with the best non-IBM run across all runs submitted to NIST. IBM has the best Average Precision at 14 out of the 17 concepts. The best Mean Average Precision of IBM system (0.263) is 34 percent better than the second best System.

5. Story Segmentation

In our news story segmentation system, we applied and extended the Maximum Entropy (ME) statistical model to effectively fuse diverse features from multiple levels and modalities, including visual, audio, and text. We have included various features such as motion, face, music/speech types, prosody, and high-level text segmentation information. The statistical fusion model is used to automatically discover relevant features contributing to the detection of story boundaries. One novel aspect of our method is the use of a feature wrapper to address different types of features, usually asynchronous, discrete, continuous, and erroneous. We also developed several novel features related to prosody. Using the news video set from the TRECVID 2003 benchmark, we demonstrate satisfactory performance and more importantly observe an interesting opportunity for further improvement.

The story boundaries defined by LDC include those of normal news stories as well as boundaries of sports and weather. Figure 5 illustrates common types of stories that can be found in broadcast news videos such as CNN. The proportion of different types in the whole collection is listed in Table 3 (row 2: percentage). Note that there is a broad range of story types with significant percentage of data. If relying on heuristic rules, i.e. the anchor segments, the performance is limited. For example, with our anchor face detection feature, the boundary detection F1 measures in ABC is 0.67 and is 0.51 in CNN with only 0.38 recall and 0.80 precision rates.

5.1. Approach

News videos from different channels usually have different production rules or dynamics. We choose to construct a model that adapts to each different channel. When dealing with videos from unknown sources, identification of the source channel can be done through logo detection or calculating model likelihood (fitness) with individual statistical station models. We propose to model the diverse production patterns and content dynamics by using statistical frameworks (i.e. ME). The assumption is that there exist consistent statistical characteristics within news video of each channel, and with adequate learning, a general model with a generic pool of computable features can

be systematically optimized to construct effective segmentation tools for each news channel. In this experiment, we take the union of shot boundaries and audio pauses as candidate points but remove duplications within 2.5-second fuzzy window. Our study showed these two sets of points account for most of the story boundaries in news [21].

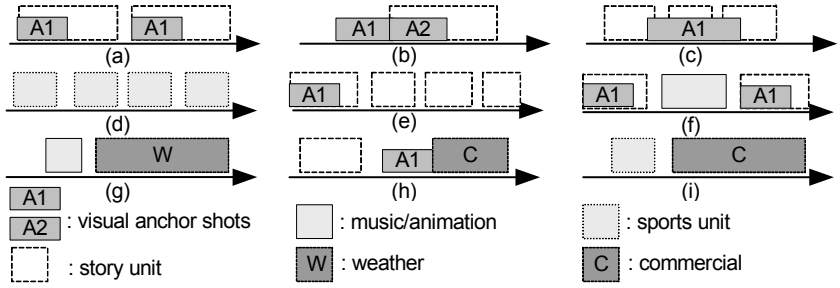


Figure 5. Common CNN story types seen in the TRECVID 2003 data set. A1 and A2 represent segments showing visual anchor persons; (a) two stories both starting with the same visual anchor; (b) the second story starts with a different visual anchor; (c) multiple stories reported in a single visual anchor shot; (d) a sports section constitutes series of briefings; (e) series of stories that do not start with anchor shots; (f) two stories that are separated by long music or animation representing the station id; (g) weather report; (h) a non-story section consists of a anchor lead-in followed by a long commercial; (i) a commercial comes right after a sports section.

5.1.1. Raw multi-modal features and feature wrapper.

We adopt the raw audio-visual features developed in [21]. They cover a wide range of features at different levels from audio, speech, and visual modalities. They include anchor face, commercial, pitch jump, significant pause, speech segment and rapidity, music/speech discrimination, motion, and ASR-based story segmentation scores [15], etc. Among them, we invented pitch jump and significant pause, prosody-related and shown to be gender and language independent **Error! Reference source not found.**[10]. The audio-visual features described above are usually diverse and asynchronous (e.g., features computed at points from surrounding time windows from different modalities). We have developed a feature wrapper to convert different types of features into a consistent form of binary features $\{g_i\}$ [21]. The feature wrapper function also contains parameters of time interval for measuring the change over time, thresholds for quantizing continuous feature values to binary ones, and the size of the observation window surrounding the candidate points. From the raw multi-modal features, we use the feature wrapper to generate a 195-dimension binary feature vector at each candidate point.

5.1.2. Maximum entropy model

The ME model [21][8] constructs an exponential log-linear function that fuses multiple binary features to approximate the posterior probability of an event (i.e., story boundary) given the audio, visual, or text data surrounding the point under examination, as shown in Equation (0.1). The construction process includes two main steps - parameter estimation and feature induction.

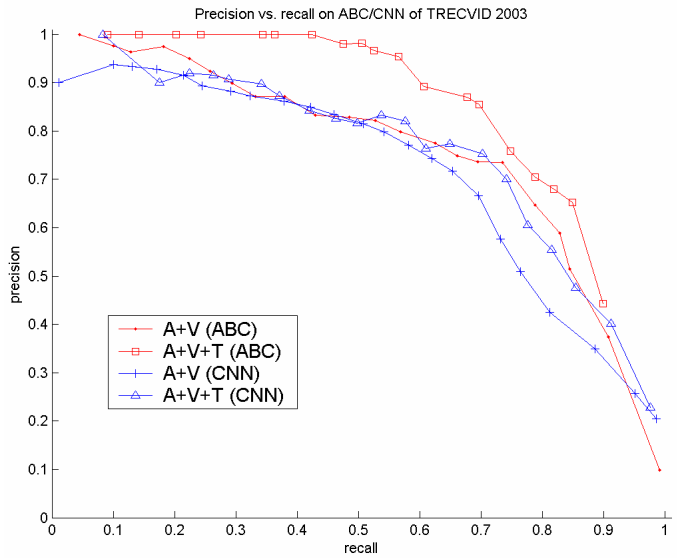


Figure 6. Precision vs. recall curves of story segmentation of ABC/CNN news with modalities "A+V" and "A+V+T".

The estimated model, a posterior probability $q_\lambda(b \mid x)$, is represented as

$$q_\lambda(b \mid x) = \frac{1}{Z_\lambda(x)} \exp\left\{\sum_i \lambda_i f_i(x,b)\right\}, \quad (0.1)$$

where $b \in \{0,1\}$ is a random variable corresponding to the presence or absence of a story boundary in the context x; $\{\lambda_i\}$ is the estimated real-valued parameter set; $\sum_i \lambda_i f_i(x,b)$ is a linear combination of

binary features; $Z_\lambda(x)$ is a normalization factor.

Meanwhile, x is the video and audio data surrounding a candidate point of story boundaries. From x we compute a set of binary features, $f_i(x,b) = 1_{\{g_i(x)=b\}} \in \{0,1\}$. $1_{\{\cdot\}}$ is an indication function; g_i is a predictor of story boundary using the i'th binary feature, generated from the feature wrapper. f_i equals 1 if the prediction of predictor g_i equals b, and is 0 otherwise.

Parameter estimation: the parameters $\{\lambda_i\}$ are estimated by minimizing the Kullback-Leibler divergence measure between the estimated model and the empirical distribution in the training set. We use an iterative process to update $\{\lambda_i\}$ till divergence is minimized.

Feature induction: from the candidate pool, a greedy induction process is used to select the feature that has the largest improvement in terms of gains or divergence reduction. The selected feature is then removed from the candidate pool. The induction process iterates with the new candidate set till the stopping criterion is reached (e.g., upper bound of the number of features or lower bound of the gain). In our experiment, we select 35 binary features.

5.1.3. Segment classification

As for segment classification, although sophisticated models can be built to capture the dynamics and features in different classes, we adopt a simple approach so far. We apply a separate commercial detector to each shot and simply compute the overlap between the computed boundary segments and the detected commercial segments. The computed segment is labeled as news if it overlaps the non-commercial portions more than a threshold; otherwise is labeled as non-news. The threshold is determined from the training set with the best argument that maximizes story classification F1 measure.

5.2. Results

5.2.1. Boundary detection performance

The story boundary detection performance, in precision vs. recall curves, on CNN and ABC videos are shown in Figure 6, where "A" means audio cues, "V" is visual cues, and "T" is text. The performance metrics are the same as those defined in TRECVID 2003. With the same modalities, the performance in ABC is better than that in CNN. It's probably due to that ABC stories are dominated by anchor segments such as type (a) in Figure 5. With suitable decision thresholds, 0.25 for CNN and 0.35 for ABC, on posterior probability $q_\lambda(1|\cdot)$ [1], for modalities "A+V+T", we could achieve F1 measure 0.76 in ABC and 0.73 in CNN; for modalities "A+V", we have 0.71 in ABC and 0.69 in CNN; with "T" alone, we derive 0.59 for both channels. After fusing text segmentation into A+V, the precision and recall are both improved even though the text feature is with real-valued scores and computed at non-speech points only, which may not coincide with those used for the audio-visual features. It is apparent that the fusion framework successfully integrates these heterogeneous features that compensate for each other. It's interesting to see that at low recall rates in ABC, the integration of text information boosts the precision almost to 1.0.

To analyze error cases, we exemplify the missed boundaries of CNN with "A+V" modalities on common types of Figure 5 in Table 3. There are high miss rates for types (b), (d), (e), and (f), where types (d) and (e) are both large groups constituting more than 36% of the whole date set when combined. It performs quite well on types (a) and (c) where the anchor face and prosody features dominate. Understanding gained by such detailed analysis of error sources will be useful for developing enhanced techniques in the future work. As for the induced binary features, the anchor face feature in a certain observation window is the most relevant; the next induced is the significant pause within the non-commercial section. More descriptions are in [21].

5.2.2. Segment classification performance

Each detected segment is further classified into news vs. non-news using the algorithm described in Section 5.1.3. We observe high accuracy of segment classification (about 0.91 in F1 measure) in both CNN and ABC. Similar accuracies are found in using different modality fusions, either A+V or A+V+T. Such invariance over modalities and channels is likely due to the consistently high accuracy of our commercial detector.

---

[1] In the official submission of TRECVID 2003, we use the decision threshold "0.5" for all setups and resulted in lower F1 measures.

5.3. Discussion

Story segmentation in news video remains a challenging issue even after years of research. We believe multi-modality fusion through effective statistical modeling and feature selection are keys to solutions. There are other perceptual features that might improve this work; for example the more precise speech rapidity measured at the phoneme level since towards the end of news stories news anchors may have the tendency to decrease their rate of speech or stretching out the last few words; in addition, the cue terms extracted from embedded text on the image might provide important hints for story boundary detection as well. One of our future directions is to explore the temporal dynamics of the news program since the statistical behaviors of features in relation to the story transition dynamics may change over time in the course of a news program.

Table 3. Percentages of missed story boundaries (row 3) of 22 CNN videos with A+V modalities. Rows 1 and 2 are the number of story boundaries and their percentages.

#	Exps./Types	a	b	c	d	e	f	g	h	i	all
1	Story Bdry. #	244	48	67	114	162	16	22	58	28	759
2	Percentage (%)	32.0	6.3	8.8	15.0	21.3	2.1	2.9	7.6	3.7	100
3	Missed Bdry. (%)	4.9	68.8	20.9	59.6	61.7	93.8	27.3	17.2	39.3	-

6. Search

We participated in the search task, submitting runs based on manual and interactive search. We explore two primary search systems, one based on speech indexing, and a second based on content-based and model-based indexing [14]. We briefly describe these two systems and discuss our different approaches for manual and interactive search.

6.1. Systems

6.1.1. Speech search system

For TRECVID2003, 3 speech-based retrieval systems were built. The baseline system was a fully automatic (i.e., queries processed without human intervention) SDR system (ASDR) based upon the ASR transcript supplied by LIMSI[23]. This was an OKAPI-based retrieval system. Simple preprocessing to remove a stop-list of words and non-informative phrases such as "Find me shot(s) of" was applied to the NIST supplied topic text to give the query words to this system. The second submitted system was a soft-boolean retrieval system (BSDR) where the user interprets the query topic and formulates a Boolean query expression which is then presented to the system. The system assigns per-word scores which are then combined to produce a document level score.

6.1.2. MPEG-7 search engine

MPEG-7 video search engine provides users with resources for building queries of video databases sequentially using multiple individual search tools. The search fusion method allows users to construct queries using techniques based on content-based retrieval (CBR), model-based retrieval (MBR), text- and speech-based retrieval and cluster navigation [13].

6.2. Manual Search

6.2.1. Speech-based retrieval

The final speech-based system (FSDR) was a fusion system based upon ASR transcript supplied by LIMSI[23] and phonetic transcripts developed in-house. It extends the speech-based system presented at TRECVID 2002. Closed-captioning and video OCR is not used. The system ranks documents based on a weighted linear combination of five separate retrieval systems. Three systems are OKAPI based (two based on documents of 100 words in length with slightly different definitions of documents and mapping from documents to shots; one indexing story-segmented versions of the transcripts such that documents do not overlap story boundaries), one soft-boolean system and one hybrid system that indexes phonetic transcripts but assigns per-word scores as in the soft-boolean system. The component system weights are determined to maximize the MAP score of the combined system on in-house queries for the development data set.

6.2.2. Content-based retrieval (CBR)

We explored methods for manual search using CBR and MBR searches. The manual approaches used query content by allowing the user to select example items and issue search based on features or models. The manual CBR system also allowed the user to form a query using models based on manual analysis of the statement of information need.

6.2.3. Automatic Multi-example CBR

Automatic visual query formulation is not a formal requirement for the TRECVID search task, but we found it highly desirable given the NIST 15 minute constraint on manual and interactive query formulation. For this reason, we developed algorithms for the selection of the best visual query examples, features to be used in image search, granularities of image search, and methods for combination of scores from multiple example image searches: each individually a challenging problem since the solution may not exploit any prior knowledge of the queries or the search set. We refer to our fully automatic CBR approach as Multi-example Content Based Retrieval (MECBR), since we automatically query content by specifying multiple visual query examples using only a single query iteration. MECBR attempts to mitigate some of the semantic limitations of traditional CBR techniques by allowing multiple query examples and thus a more accurate modeling of the user's information need. It attempts to minimize the burden on the user, as compared to relevance feedback methods, by eliminating the need for user feedback and limiting all interaction—if any—into a single query specification step. It also differs from relevance feedback (RF) methods in that MECBR usually involves the execution and combination of multiple simple queries rather than the continuous refinement of a single query, as in typical RF methods. MECBR also works well with a small number of examples, where traditional statistical modeling methods typically fail due to lack of sufficient training data. The design of MECBR therefore positions it as a lightweight alternative for modeling of low-level and mid-level semantic topics, including semantically/visually diverse topics as well as rare topics with few training examples (e.g., see [2][3]).

An important aspect of the MECBR approach is the underlying content-based features used for computing image similarity. The features we investigated included 166-dimensional HSV color correlograms for capturing of visual similarity and 46-dimensional semantic model vectors for capturing of semantic similarity. The model vector for each keyframe is constructed by analyzing the keyframes using 46 semantic concept models to detect the presence/absence of these 46 concepts. Each such detection results in a confidence measure which is then used to construct a feature vector in the semantic space of the 46 concepts. Concepts used for constructing the model vector include 46 of the most frequently occurring concepts in the NIST TREC Video Development Corpus. The models used for analyzing each concept's presence in a keyframe are based on uni-modal Support Vector Machine classifiers constructed in a manner described in Section 4.4.2. The model vector space is thus a nonlinear mapping of the original visual features, including color, texture, shape and edges, using supervision of explicit semantic concepts.

Our original MECBR formulation does not require prior training or use feedback but it does require the user to specify one or more query examples and a fusion method to be used in combining these per-example query results. This requirement must be removed if we are to use it as our fully automatic image retrieval system. It is a challenging problem to select the best query examples and fusion methods fully automatically. Our solution is as follows. We use all provided example images—including all I-frames from given relevant video clips—but, to reduce sensitivity to noise and outliers, we categorize these examples into visually/semantically coherent categories (or clusters). We then perform multiple image retrievals and formulate the equivalence of complex Boolean queries in the visual domain—using fuzzy AND logic for fusion within categories and fuzzy OR logic for fusion across categories. We treat each category as equally important in retrieval, irrespective of its size, so that visually/semantically distinct examples have an equal chance of impacting performance. Within a category, though, the importance of an example is defined to be inversely proportional to its distance from the category centroid. The idea is to boost importance of features that occur frequently across category examples while diminishing importance of rare features or noise in the categories. The categorization itself is performed by clustering in the visual or semantic feature domain using a maximum cluster radius threshold and resulting in a variable number of clusters depending on the homogeneity of the query topic examples. To improve robustness of clustering with few examples, we use an iterative clustering algorithm which greedily selects new clusters that are maximally apart from the currently selected clusters at each iteration. Clustering stops when no new clusters can be formed given the specified cluster radius threshold, resulting into the set of most distinct clusters with a radius smaller than the given distance threshold.

In order to formulate and execute the final query, each example image is used to form an atomic query in both the visual and the semantic feature space. Each atomic query is then answered by ranking all candidate images with respect to their Euclidean distance to the query image in the corresponding feature space. For the semantic features,

this is done at the granularity of whole images only since regional concepts are already accounted for in the SVM concept detection phase. For visual features, however, similarity is computed for the whole image, as well as at a sub-image granularity level, using a fixed 5-region layout (4 equal corner regions and an overlapping center region of the same size). Global and local similarity is then averaged to form a single visual similarity score for each image. Similarly, the visual and semantic similarity scores are then combined to form an overall similarity score for each keyframe with respect to a single query image example. Similarity scores for entire clusters are then generated by weighted averaging of individual per-example results within the cluster, using the corresponding example importance weights. All cluster scores are then aggregated, this time using max as the score aggregation function in order to simulate logical OR behavior. The final score for each candidate image is therefore derived after fusion at multiple levels, including visual and semantic features (score averaging), global and regional image granularities (score averaging), examples within a cluster (weighted score averaging), and clusters within a given topic (max score). The resulting fully automatic visual/semantic run performed on par with the human expert-formulated visual/semantic run, attesting to the promise of the approach. This run was also used in generating our best multi-modal manual and interactive runs (see Sections 6.2.4 and 6.3.3), which outperformed the best uni-modal runs by 20% and 40%, respectively.

6.2.4. Multimodal retrieval systems

The first manual multimodal system formed a query-dependent linear weighted combination of two independent unimodal systems (speech-based and image content-based): specifically FSDR and MECBR systems. This was in part a consequence of NIST test specifications of 15 minute time-limit on query formulation. Ideally, the ISDR and ICBR interactive runs could be combined. However, this would violate the 15 minutes limit, since each of the two runs used 15 minutes user time, independently. The weights to combine these two systems are chosen by the users based on their experience to predict which modality is expected to perform better for particular queries. This modality receives 70% of the weight (30% for the other modality) and the result set is re-ranked using this weighed combination of scores. Note that while the combination weights are query-dependent, the user has no knowledge of the actual performance of the component systems on the particular query. Further analysis of the user-prediction (optimality or lack thereof) will appear in future work.

The second multimodal system formed query-independent linear combination of scores from two retrieval systems (speech-based and image content-based): specifically FSDR and a content-based run. The speech-based system was run first and the top 1000 shots are retained. A constrained global or regional content-based search is then performed using a user-selected “best” query image, region(s) and low-level features. The final score assigned is a weighted combination of the FSDR score and the constrained content-based search score. Query-independent per-modality weights were chosen to maximize MAP score for in-house queries on development data.

6.3. Interactive Search

An interactive IR system may provide combined searching and browsing, query refinement, relevance feedback, and various static and dynamic visualizations [9]. An overview of interactive video retrieval systems, including design methodologies and a features comparison between different video retrieval systems is provided in [11]. Our interactive search work at TRECVID 2003 consists of two independent retrieval systems, one for Content-Based (ICBR) and one for speech-based (ISDR) retrieval. A third, fused run was submitted, in which interactive and automatic multimodal retrieval results are combined. This last run was ranked the best of all ten IBM search runs. As is consistently evident from three years of TRECVID Search task, MAP of interactive search runs is significantly higher than manual search. Interestingly, 11 of the 25 topics were better retrieved in one of our manual runs than in any of our interactive runs. Thus these runs were performed by different users, this observation suggests that we can still improve our interactive systems to better exploit the full potential of a user in the loop.

6.3.1. Speech-based retrieval

The purpose of our ISDR system is twofold: query refinement and shots elevation. In order to refine the Boolean-like text query, the user starts with an initial query, browses the result table, possibly listens to some of the retrieved video shots, and then modifies the query and repeats the process. After several such iterations, the refined query in its final form. Next the user may mark individual shots in the obtained results list as “relevant" or “irrelevant". In this part, the main challenge is to build an interface that provides efficient video browsing and fast results marking. Upon saving the final marked list, all shots marked as relevant are elevated to the top of the list, the ones marked

irrelevant are pushed to the bottom of the list, and the rest, unmarked shots, are left sorted by their original retrieval scores. This re-ranked list makes the final ISDR run result set for the search topic.

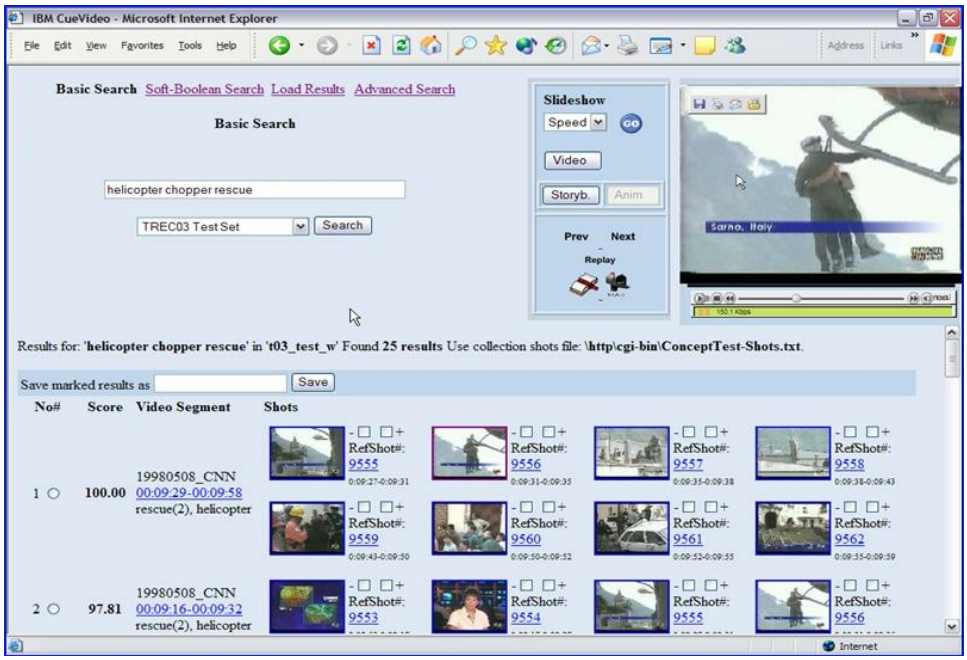


Figure 7. The Interactive SDR GUI is built of a query region, a results list region and an enhanced video browser with multiple synchronized views.

The major distinction of ISDR from ICBR is the user's frequent need to play some of the retrieved shots and listen to the spoken words. This is in contrast to ICBR, where audio and speech are mostly irrelevant and visual browsing of thumbnails might be just enough as the feedback for most topics. Visual browsing is more time efficient compared to browsing of audio and speech. Multiple thumbnails may be rendered in a single web page and browsed in a glance. When it comes to browsing of speech, however, only a single source can be played at a time, and it requires much more time to play than to glance at a static image. To be efficient, our ISDR system (shown in Figure 6), have the following properties:

1. Fast response to user's input. Short search time, quick visual rendering of the results with thumbnails and the relevant words from the speech.
2. Rapid video playback access to any selected video segment/shot.
3. Easy video navigation including pause, play and replay of a selected segment.
4. Session awareness - to support easy query modification
5. Multiple synchronized views, including video, storyboard, and slide shows with fast audio.

System architecture and implementation details may be found in [1]. The 25 search topics were split between seven users, 3-4 topics per user, in a pseudo-random order. Six of the users were located about 2500 miles away from the ISDR server. The usage instructions were given over the phone and were summarized in a single page email. No user training session was conducted, thus minimal training could have helped to improve interactive search performance. The Interactive SDR system achieved a MAP of 0.1434. This is compared to a MAP of 0.0846 achieved by the same SDR engine in a Manual Search run, without interaction with the test set. The improvement is credited to the two interaction steps, namely the query refinement and the elevation of individual relevant shots and suppression of irrelevant ones. The user was allowed to up to 15 minutes interaction time, and has to decide how to split the time between the two steps. The ISDR run produced the best IBM AP on four topics: 106 – Unknown Soldier Tomb, 107 - Rocket launch, 108 - Mercedes Logo, and 113 - Snow covered mountains. On the other hand, no correct matches were found on topics 118 - Mark Souder and 121 - coffee mug. Topics 102 - baseball and 120 - Dow Jones were better detected by the Manual SDR system than by the interactive one. Since those are identical retrieval systems, the reason is only due to different queries composed by different users, on different data sets (development vs. test set). These results suggest that there is more potential for improvement in the Interactive system.

6.3.2. Content-based retrieval (CBR)

The interactive CBR system provides user controls for fusing multiple searches using different search methods using different normalization and combination methods and aggregation functions as shown in Figure 8. The objective is to provide the user with the greatest flexibility and power for composing and expressing complex queries [13].

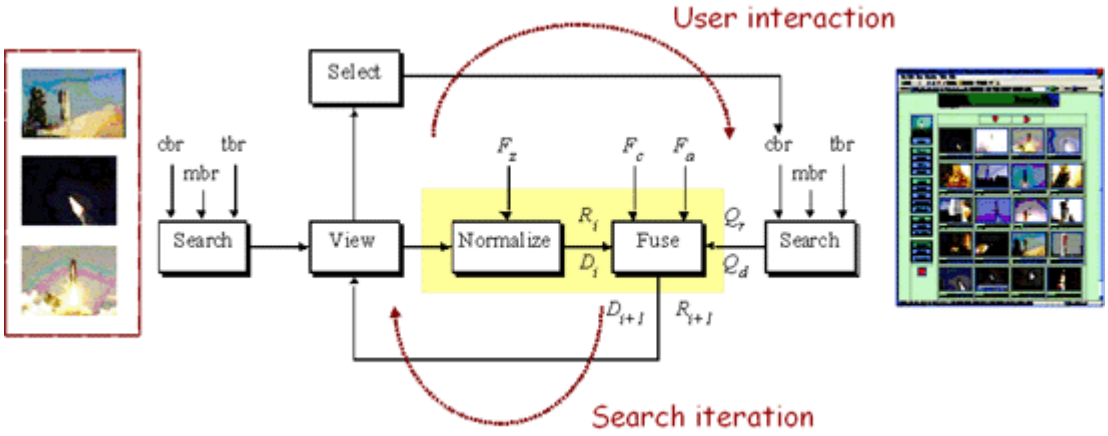


Figure 8. Overview of search fusion method for interactive content-based retrieval.

6.3.3. Interactive Fusion

For our interactive multi-modal runs, we chose to allow the user to spend all of the allowed 15 minutes interactively exploring only one of the two modalities (visual or speech) for the given query topic. We then combined these interactive results with the results of a fully automatic (zero user effort) search in the complementary modality, using a late fusion aggregation strategy. In other words, because we had generated fully automatic runs in both the speech-based and content-based modalities, we could use the entire allotted time exploring the more promising modality more effectively, and to then improve results by combining them with an automatic run from the other modality (at a zero cost with respect to the total user time spent for a given query). With the final weight determination strategy, the user has full knowledge of both the query topic and the performance of the two independent systems on that topic. Based on that knowledge the user determines the mixing weights A and B for the two modalities, where A+B=10. The multimodal run is computed by interlacing the two result sets (sorted by relevance) so that the aggregated run takes the top A items from the first set followed by the top B distinct items from the second set, and so on; specifically, the interactive run for the higher-weighted modality is interlaced with the automatic run for the lower-weighted modality. This system resulted in our highest overall performance for the Search task and improved upon the best uni-modal system by 40% with respect to Mean Average Precision over the query topics.

7. Summary

In this paper we described our participation in the NIST TRECVID-2003 evaluation and discussed our approaches and results in the four tasks of the benchmark including shot boundary detection, high-level feature detection, story segmentation, and search.

8. Acknowledgments

This work could not happen without the help and support of several organizations and many of our colleagues. We would like to express our deepest thanks to Paul Over and NIST for organizing TRECVID 2003 and to Alan Smeaton and Wessel Kraaij for coordinating it. Special thanks to the Linguistic Data Consortium (LDC) and to C-SPAN for providing the video data and metadata for the TRECVID benchmark. We like to express our thanks to Jean-Luc Gauvain, LIMSI, for preparing automatic speech transcripts and making them available to all TRECVID participants. Thanks to our university colleagues; Shan Sachedv (MIT), Javier Ruiz-del-Solar, Alex Jaimes, Dinko Yaksic and Rodrigo Verschae (Univ. of Chile), and to all our IBM colleagues who contributed from their time and research tools; Stanley Chen, Chitra Dorai, Martin Franz, Christian Lang, Ying Li, Larry Sansone, and our

volunteers from the Human Language Technologies (HLT) group. Last but not least, we are very grateful to all the 111 researchers around the world who took part in the collaborative annotation of the entire development set.

9. References

[1] A. Amir, S. Srinivasan, and D. Ponceleon. "Efficient video browsing using multiple synchronized views." In Azriel Rosenfeld, David Doermann, and Daniel DeMenthon, editors, *Video Mining*. Kluwer Academic Publishers, Boston, USA, 2003.

[2] A. Natsev, J. R. Smith, "Active Selection for Multi-Example Querying by Content," *Proc. IEEE Intl. Conf. on Multimedia and Expo (ICME)*, Baltimore, MD, July, 2003.

[3] A. Natsev, M. Naphade, J. R. Smith, "Exploring Semantic Dependencies for Scalable Concept Detection," *Proc. IEEE Intl. Conf. on Image Processing (ICIP)*, Barcelona, ES, Sept., 2003.

[4] B. L. Tseng, C.-Y. Lin, M. Naphade, A. Natsev and J. R. Smith, "Normalized Classifier Fusion for Semantic Visual Concept Detection," *IEEE Intl. Conf. on Image Processing*, Barcelona, Sep. 2003.

[5] C.-Y. Lin, B. L. Tseng and J. R. Smith, "Video Collaborative Annotation Forum: Establishing Ground-Truth Labels on Large Multimedia Datasets", *Proc. of NIST TREC Video 2003*.

[6] C.-Y. Lin, B. L. Tseng, and J. R. Smith. "VideoAnnEx: IBM MPEG-7 Annotation Tool for Multimedia Indexing and Concept Learning." *Proc. IEEE Intl. Conf. on Multimedia and Expo (ICME)*, Baltimore, MD, July, 2003.

[7] C.-Y. Lin, B. L. Tseng, M. Naphade, A. Natsev, J. R. Smith, "VideoAL: A end-to-end MPEG-7 video automatic labeling system," *Proc. IEEE Intl. Conf. on Image Processing (ICIP)*, Barcelona, ES, Sept., 2003.

[8] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation, " *Machine Learning*, vol. 34, special issue on Natural Language Learning, pp. 177-210, 1999.

[9] E. D. Barraclough77. On-line searching in information retrieval, *Journal of Documentation*, 33:220-238, 1977.

[10] E. Shriberg, A. Stolcke, D. Hakkani-Tur, and G. Tur, F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498--519, 2001.

[11] H. Lee and A. F. Smeaton. Designing the user interface for the Fischlar digital video library. *Journal of Digital information*, Special Issue on Interactivity in Digital Libraries, Vol.2:4, Article No. 103, 2002-05-21, 2002.

[12] H. Nock, W. Adams, G. Iyengar, C-Y Lin, M. R. Naphade, C. Neti, B. Tseng, J. R. Smith, "User-trainable Video Annotation using Multimodal Cues", *Proc. SIGIR 2003*

[13] J. R. Smith, A. Jaimes, C.-Y. Lin, M. Naphade, A. Natsev, B. L. Tseng, "Interactive Search Fusion Methods for Video Database Retrieval," *Proc. IEEE Intl. Conf. on Image Processing (ICIP)*, Barcelona, ES, Sept., 2003.

[14] J. R. Smith, M. Naphade, A. Natsev, "Multimedia Semantic Indexing using Model Vectors," *Proc. IEEE Intl. Conf. on Multimedia and Expo (ICME)*, Baltimore, MD, July, 2003.

[15] M. Franz, J. S. McCarley, S. Roukos, T. Ward, and W.-J. Zhu, "Segmentation and detection at IBM: Hybrid statistical models and two-tiered clustering broadcast news domain," in *Proc. of TDT-3 Workshop*, 2000.

[16] M. Naphade and J. R. Smith, "A Hybrid Framework for Detecting the Semantics of Concepts and Context", 2nd *Intl. Conf. on Image and Video Retrieval*, pp. 196-205, Urbana, IL, June 2003

[17] M. Naphade and J. R. Smith, "Learning Visual Models of Semantic Concepts", *IEEE International Conference on Image Processing*, Barcelona 2003

[18] M. Naphade, I. Kozintsev and T. Huang, "A Factor Graph Framework for Semantic Video Indexing", *IEEE Transactions on Circuits and Systems for Video Technology* Jan 2002.

[19] M. Naphade, J. R. Smith, "Learning Regional Semantic Concepts from Incomplete Annotations," *Proc. IEEE Intl. Conf. on Image Processing (ICIP)*, Barcelona, ES, Sept., 2003.

[20] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*, chapter 10: User Interfaces and Visualization, pages 257{323. Addison Wesley, Reading, USA, 1999.

[21] W. Hsu, S.-F. Chang, C.-W. Huang, L. Kennedy, C.-Y. Lin, and G. Iyengar, "Discovery and fusion of salient multi-modal features towards news story segmentation," in *IS&T/SPIE Electronic Imaging*, San Jose, CA, 2004.

[22] W.H. Adams, A. Amir, C. Dorai, S. Ghosal, G. Iyengar, A. Jaimes, C. Lang, C.-Y.Lin, A. Natsev, C. Neti, H. J. Nock, H. Permuter, R. Singh, J. R. Smith, S. Srinivasan, B. L. Tseng, AT Varadaraju, D. Zhang,"IBM Research TREC-2002 Video Retrieval System," *NIST Text Retrieval Conference (TREC-2002),* Nov., 2002.

[23] J.L. Gauvain, L. Lamel, and G. Adda. The LIMSI Broadcast News Transcription System Speech Communication, 37(1-2):89-108, 2002. ftp://tlp.limsi.fr/public/spcH4_limsi.ps.Z

[24] T. Joachims, "Making large-scale SVM learning practical", Support Vector Learning, ed. B. Schölkopf, C. Burges and A. Smola, MIT Press, 1999