Dublin City University Video Track Experiments for TREC 2002

P. Browne, C. Czirjek, C. Gurrin, R. Jarina, H. Lee, K. Mc Donald, A. F. Smeaton and J. Ye

Centre for Digital Video Processing, Dublin City University, Glasnevin, Dublin 9, Ireland
Alan.Smeaton@computing.dcu.ie

Abstract

Dublin City University participated in the Feature Extraction task and the Search task of the TREC-2002 Video Track. In the Feature Extraction task, we submitted 3 features: Face, Speech, and Music. In the Search task, we developed an interactive video retrieval system, which incorporated the 40 hours of the video search test collection and supported user searching using our own feature extraction data along with the donated feature data and ASR transcript from other Video Track groups. This video retrieval system allows a user to specify a query based on the 10 features and ASR transcript, and the query result is a ranked list of videos that can be further browsed at the shot level. To evaluate the usefulness of the feature-based query, we have developed another system interface that provides only ASR transcript-based querying, and we conducted an experiment with 12 test users to compare these 2 systems. Results were submitted to NIST and we are currently conducting further analysis of user performance with these 2 systems.

1. Introduction

This year Dublin City University took part in two of the three tasks in the Video Track: Feature extraction task and Search task. In Section 2 we present the Feature extraction task that we conducted (Face, Speech, and Music), and the methods used for each feature, and our results. In Section 3 we present the Search task – specifically the interactive video retrieval system that we developed for the task and the experiment procedures and our results. The system is a variation of the Físchlár Digital Video System with an XML-based architecture that uses an MPEG-7 compliant video description. The system provides a web-based user interface from which a user can compose a query based on all the 10 features and/or the ASR transcript text. We used the system in the interactive search task with 12 test users who each conducted searches for the 25 topics provided by the Video Track.

2. Feature Extraction Task

Ten features were listed for the feature extraction task and we extracted three of the features ourselves: Speech, Instrumental Sound and Face.

2.1 Speech Extraction

The task here is to recognise a shot as having a human voice uttering words. In our approach, speech characteristics were derived from the volume (energy) contour of the frequency-limited audio signal. There are some properties that distinguish speech from other signals. Roughly, speech exhibits an alternating sequence of 3 kinds of sounds that have different acoustic properties: i) *vowels and vowel-like sounds* – longer tonal quasi-periodic segments with high energy, which is concentrated in lower frequencies; ii) *fricative consonants* – noise-like short segments with lower volume and spectral energy distributed more toward the high frequencies; iii) *stop consonants* – short silent segments followed by a very short transition noise pulse. These three kinds of sounds alternate and form the regular syllabic structure of speech and therefore strong temporal variations in the amplitude of speech signals are observed.

Our speech detector does not use an audio signal waveform as the input data, rather it utilises information taken directly from an MPEG-1 audio encoded bitstream. Thus a time-consuming decoding process is not required, and in addition information from audio signal analysis (e.g. subband filtering, volume estimation) already stored in the MPEG encoded bitstream, is utilized. The MPEG audio layer-II frame consists of 1,152 samples: 3 groups of 12 samples from each of 32 subbands. A group of 12 samples in each subband gets a bit allocation and, if this is not zero, a scalefactor. Scalefactors are weights that rescale samples so that they fully use the range of the quantizer. The encoder uses a different scalefactor for each of the three groups of 12 samples only if necessary. By definition the scalefactors carry information about the maximum level of the signal in each subband. Thus, the volume contour of the overall audio signal can be estimated by the summation of the scalefactors over all subbands. In the case of a frequency-limited signal, this summation is done only over given subbands.

2.1.1 Procedure for Speech Detection

Our approach was based on the measurement of the duration and the rate of the energy peaks of the audio signal. The method was first introduced in [1] where theoretical background and results of preliminary studies can be found. For the TREC task, the method had to be slightly modified. By trial examination of various parts of video recordings from the TREC *Feature Development Collection*, it was decided that at least 7 of the low frequency subbands must be included in the processing. In some cases, the first subband (frequencies up to 0.7 kHz) was excluded from analysis. The procedure of signal analysis and processing for speech detection is described below.

- Each video file was demultiplexed, and the MPEG-1 audio layer II bitstreams were stored in separate files (MP2 files). Then, only the scalefactors of the first 7 subbands were extracted from the MP2 files.
- First, silence detection was carried out. An energy level of the signal was determined by the superposition of all relevant scalefactors. The frames in which the level was below the threshold, were assigned as silent frames.
- In the case of speech detection, the envelope of the band-limited signal was estimated by summing relevant scalefactors only from the 2^{nd} to the 7^{th} subbands. This procedure was followed by a 5^{th} order median filtering to avoid rapid random changes in the amplitude.
- For analysis, a sliding window was used with a window length of 3.9 seconds and a 1.3 seconds shift (i.e. 2/3 overlap).
- Energy peaks were extracted by a simple thresholding procedure. Two low-level features were chosen for speech detection. These are: i) *Lm* – the duration of the widest peak within the analysis window (segment); ii) *R* – the rate of peaks (number of peaks in the analysis window). Each segment was assigned to speech or non-speech by using a simple rule-based decision procedure. This process has been discussed in greater detail in [1]. All the MP2 audio frames corresponding to an analysed segment were given a relevance value of '1' in the case of a speech segment, and the value '0' otherwise.
- The silent parts of the signal longer than 1.5 seconds were assigned as non-speech signal.
- Final speech feature measures for the standard video shots were determined by averaging the relevance values over all the audio frames within each video shot.

2.1.2 Evaluation and Test Results

For evaluation, we submitted the top 1,000 standard video shots ranked according to the highest possibility of detecting the speech feature. The results of our runs are summarized in Table 1.

	Our Results	Maximum	Median
Average precision	0.710	0.721	0.656
Precision at 100 results	1.000	1.000	0.980
Precision at 1000 results	0.987	0.997	0.944

Table 1. Speech test results (compared with maximum and median values)

2.2 Instrumtental Sound Extraction

This feature characterises a sound produced by one or more musical instruments. Henceforth this feature is referred to as the *music feature*. The music feature detection task is a much more challenging task than the speech detection task. Unlike speech, musical sounds are very difficult to define due to their great variety and uncertain nature. However, musical signals have some unique characteristics, which may help to discriminate them from other sounds. Music tends to be composed of a multiplicity of tones, each with its own distribution of higher harmonics. The energy contour has usually a much smaller number of "peaks" and "valleys" and it shows either very little change over a period of several seconds (e.g. classical music) or strong long term periodicity due to exact rhythm (e.g. dance music).

For this TREC task we developed a method that is an extended version of the method we already used for speech detection (see section 2.1). Two other low-level features were incorporated into the system to improve discrimination between musical sounds and other environmental sounds. They are: rhythm and harmonicity. We believe that most of the sounds produced by instrumental music have harmonic structure of spectra unlike noise-like environmental sounds. The importance of rhythm detection has been recently discussed in [2].

2.2.1 Procedure for Music Detection

The first two features *Lm* and *R* (duration and rate of the energy peaks) were computed in the same way as in section 2.1.1. In addition, the rhythm (or pulse) metric *Pm* and harmonic ratio H were computed.

- Similar to [2], the rhythm metric is expressed by the following procedure.
  For each of the first 7 subbands, the normalized autocorrelation function $\overline{R}$ were computed.

$$\overline{R}_k(t)=R_k(t)/R_k(0),\quad R_k(t)=\sum_n e_k(n)\cdot e_k(n+t),$$

where $e_k(n)$ means the subband energy contour (or envelope) in the k-th subband, without its DC component. The subband energies were estimated directly from the scalefactors of the MPEG-1 layer II bitsream. For rhythm analysis, a sliding window with a 4/5 overlap was used. We searched $\overline{R}$ within the analysed window over the interval corresponding to time $t = 0.2$ - 1.75 seconds to find peaks. We set p(j) to the value of the highest peak in the j-th subband. Then we defined the feature rhythm metric P_m as

$$P_m = \max\{p(k)\},\quad k=1,2,...7,\qquad 0<P_m<1$$

The higher the value of P_m, the greater amount of rhythmicality in the signal.

- The harmonicity ratio defines the degree of harmonicity of an audio signal. We computed it in accordance with the MPEG-7 description schema [3]. By definition, the harmonicity ratio is the ratio of harmonic power to total power. It was computed by the following procedure:

At first, comb filtering is applied

$$r(k)=\sum_j s(j)s(j-k)\Bigg/\left(\sum_j s(j)^2 * \sum_j s(j-k)^2\right)^{0.5}$$

where s is sequence of PCM samples of the band-limited signal. Only the 2nd subband was used for the computation. Thus the sampling frequency was f_2=44.1kHz / 32. Index k was changed up to the value corresponding to the maximum expected fundamental period (around 20 ms). The Harmonicity ratio H was determined as the maximum value of $r(k)$ for each frame. $H = 1$ for a purely periodic signal, and it will be close to 0 for white noise.

- These four low-level features *Lm, R, Pm* and *H* were used as inputs for a heuristic rule-based classifier. The relevance for each analysed segment was computed as a weighted sum of these features. The weights and thresholds for the classifier were determined by trial and error examination of various parts of video recordings from the TREC *Feature Development Collection*. Similarly as in the case of speech detection, silence detection was performed. For the silent parts, which were longer than 1.5 seconds, the music relevance was set to zero.
- Final music/instrumental sound feature measures for the standard video shots were determined by averaging the relevance scores over all the audio frames corresponding to the given video shot.

2.2.2 Evaluation and Test Results

For evaluation, we submitted the top 300 standard video shots ranked according to the highest possibility of detecting the speech feature. The results are summarised in Table 2. We reach the highest precision at 100 results among TREC participants, but the precision at 1,000 results and the average precision are very low because we submitted only 300 results for evaluation. Since we identified much more relevant shots than we submitted for judgment, we have re-calculated precision and average precision for our 1,000 top ranked shots and the new results are also shown marked with * in Table 2. The new average precision is 0.705 which shows that our method performs very well.

	Our results	Maximum	Median
Average precision (official) Average precision (unofficial)	0.222 0.705 *	0.637 0.705 *	0.347 0.474 *
Precision at 100 results	0.970	0.970	0.845
Precision at 1000 results (official) Precision at 1000 results (unofficial)	0.281 0.650 *	0.877 0.877 *	0.667 0.679 *

Table 2. Instrumental sound test results

2.3 Face Extraction

As presented in [4], the colour of human skin falls into a relatively narrow band of the colour space. Many colour models have been used in pre-processing the input image, in order to locate potential human presence. We know [5] that normalized RGB, YUV, HSV, CIEL etc. can be used for this purpose. In this task, we decided to detect skin-like pixels using a similar approach to [6], updating the filtering technique based on the available *Feature Development Collection*.

2.3.1 Procedure for Face Detection

Due to the binary nature of classification, the output skin-mask will be populated with isolated skin-like pixels, i.e. noise. In order to address this undesirable effect, we applied a morphological open-close filtering. After this operation, we expect to obtain homogeneous areas of connected pixels. Having the skin-map, we want to group together connected pixel areas into regions. Therefore a connected component labelling was performed, which gave the number of regions used in further processing. Even applying morphological filtering to the skin-map, regions with a small number of pixels may occur. To reduce the number of false candidate regions, areas with the number of pixels less than $N = 625$ were ignored. We have chosen this threshold based on the assumption that no face could be detected by this method having a size smaller than 25x25 pixels. Horizontal and vertical strips, which are less likely to contain a human face, were also ignored. These regions were detected by having a huge difference between width and height, with the requirement that the smaller dimension does not exceed 25 pixels.

Assuming that the human face has an approximately elliptical shape, for each connected component (region) the best-fit ellipse was calculated based on moments [7]. Unfortunately, many other objects in a visual scene have the same colour characteristics as the human skin, or other object(s) are merged with the face (i.e. hands, background wall, etc.). An intermediate step in the processing chain consists of an iterative partitioning of regions having “irregular” shape. This means breaking a region *S* into component convex sub-regions *Sn*, *n* being the number of sub-regions, by applying K-means clustering.

The detection task is based on principal component analysis of the remaining skin patches. Given a collection of test images, we constructed a face space for discriminating the remaining candidate regions. The measure of “faceness” of the input sample relies on the reconstruction error, expressed as the difference between the input image and its reconstruction using only the M eigenvectors corresponding to the highest eigenvalues:

$$\varepsilon^2 = \| x - \bar{x} \|^2 \text{ (DFFS)}$$

The distance from face space (DFFS) indicates how well the test image can be approximated by the most significant eigenvectors spanning the eigenspace. The distance between the projected input image and the mean face image in the feature space is given by the norm of the principal component vector. Since the variance of a principal component vector y_I is given by its associated eigenvalue λ_i, the squared Mahalanobis distance measure d^2 gives a measure of the difference between the projection of the test image and the mean face image of the training set {x}:

$$d^2 = \sum_{i=1}^{M} \frac{y_i}{\lambda_i}$$

where y_I are the projection coefficients and λ_i are the associated eigenvalues. Therefore d^2 can be expressed as the distance in face space (DIFS). Given these two distances a combined error criterion was used:

$$e = d^2 + c\varepsilon^2 .$$

where $e \in [0,1]$, and c is a suitable constant value. As we work with digital video, a confidence measure is attached to each continuous video shot, meaning the level of certainty that a face occurs. Because of temporal constraints, the above algorithm processes each 10-th frame in sequence. The confidence measure for a shot is expressed as the average confidence value of each processed frame within the shot.

2.3.2 Evaluation and Test Results

We submitted for evaluation the highest ranked 300 shots and our results are summarized in Table 3. Within the first 100 shots, half of what we detected was correct but we have not yet carried out an examination of our top 1,000 results.

	Our Results	Maximum	Median
Average precision	0.154	0.613	0.166
Precision at 100 results	0.530	0.990	0.540
Precision at 1000 results	0.114	0.312	0.221

Table 3. Face detection results

3. Interactive Search Task

For the Search task, we conducted an interactive search experiment with test users. For this we developed an interactive video searching/browsing system which is a variation of the Físchlár system, and conducted a lab experiment using the system with 12 test users. The hypothesis we were testing was that ASR + features searching outperformed ASR-only searching in our controlled environment.

3.1 System Description

The system we used for the search task is a variation of the Físchlár Digital Video System [8], an online video system which has been operational for 3 years within the University campus and which we used for the interactive search task in the previous year's Video Track where we compared 3 different keyframe browsers. Currently the Físchlár system has a XML-based architecture and uses MPEG-7 compliant video description internally. While having the same underlying architecture as the Físchlár system, the system we tailored for this year's search task is more sophisticated in its search mechanism and user interface, as it provides various query methods for users based on the feature extraction data, some of which is our own (Face, Speech, Music) as well as donated features namely Indoor, Outdoor, People, Landscape and Text Overlay from IBM, and Monologue and Cityscape from Microsoft Research Asia. The system also allows the users to execute text queries over the test collection, based on the donated Automatic Speech Recognition (ASR) transcript provided by LIMSI. This transcript used an American English broadcast news transcription system and is described in [11].

3.1.1 System Architecture

Figure 1 shows the components of the Físchlár system. The system uses an internal XML description as its core element (in the centre of Figure 1). When a user submits a query via the web-based interface, the web application processes it and sends the query detail to the logic element governing the search engine (see Section 3.1.2). The search engine sends back the retrieved results with relevance scores to the XML generator, which generates the necessary XML descriptions dynamically, to be transformed by appropriate XSL stylesheets to render HTML and SVG for display back on the user's web browser.

The queries that a user generates can be composed of any, some, or all of the following elements:

- *Feature listing of the required features*, there were ten features in all (excluding ASR transcript) and the user could select any of these features for inclusion in the query. However, there were some interface restrictions placed on users, for example, a user could not specify in a query that shots *be both* Indoor and Outdoor.
- *Query text*, which would be matched against the ASR transcript. While the system supported querying based on features alone, our findings indicated that all users relied on ASR text when constructing queries.
- *An identifier of the video within which to search*, if the query was at the shot level. Our system supported both searching for videos and searching for shots within a particular video, hence the support for specifying a shot identifier within certain queries.

3.1.2 Retrieval and Weighting Scheme

In order to support search and retrieval over the video data we developed a search server, which was designed to support both ASR-only querying and ASR + feature querying for both the shots and the videos as a whole (the lower half of figure 1). Each user's search session is essentially a two-phase process. The first phase was to generate a ranked list of videos in response to a user query, where each of the 176 videos were scored and ranked before being returned in decreasing rank order to the user. The user could then select one of the videos (usually one of the higher ranked) for shot-level examination, which was the second phase. Shot-level examination results in the search server producing a ranked listing of shots from within the selected video that match the user's query, the same query that originally generated the ranked list of videos.

Our ranking technique was developed without using the TREC topics (no training data) and thus it was not developed specifically to provide high retrieval performance on this particular corpus and associated queries. With the release of the relevance judgements we are now in a position to develop and test alternative techniques for ranking both videos and shots using evidence from ASR text search as well as from the extracted features.

The ASR transcripts for each shot (donated by LIMSI) were pre-processed to remove stopwords and then stemmed using Porter's algorithm. When a user submits search term(s) as part of a query, these search terms undergo the same process. Each shot was represented by the ASR transcript text associated with the particular shot while each video was represented by the combination of all ASR text associated with all the shots that comprise the particular video. This required the utilisation of two conventional (text-only) search engines based on BM25 with the following parameter values; advl = 900, b = 0.75, k1 = 1.2 and k3 = 1000 which were set according to the best performance achieved on the WT2g collection from TREC-8 [9]. The scores for each query were normalised to be in the range [0..1] to allow for easier combination with the feature scores. We note that additional experimentation would be necessary to tune BM25 parameters to best-fit ASR content.

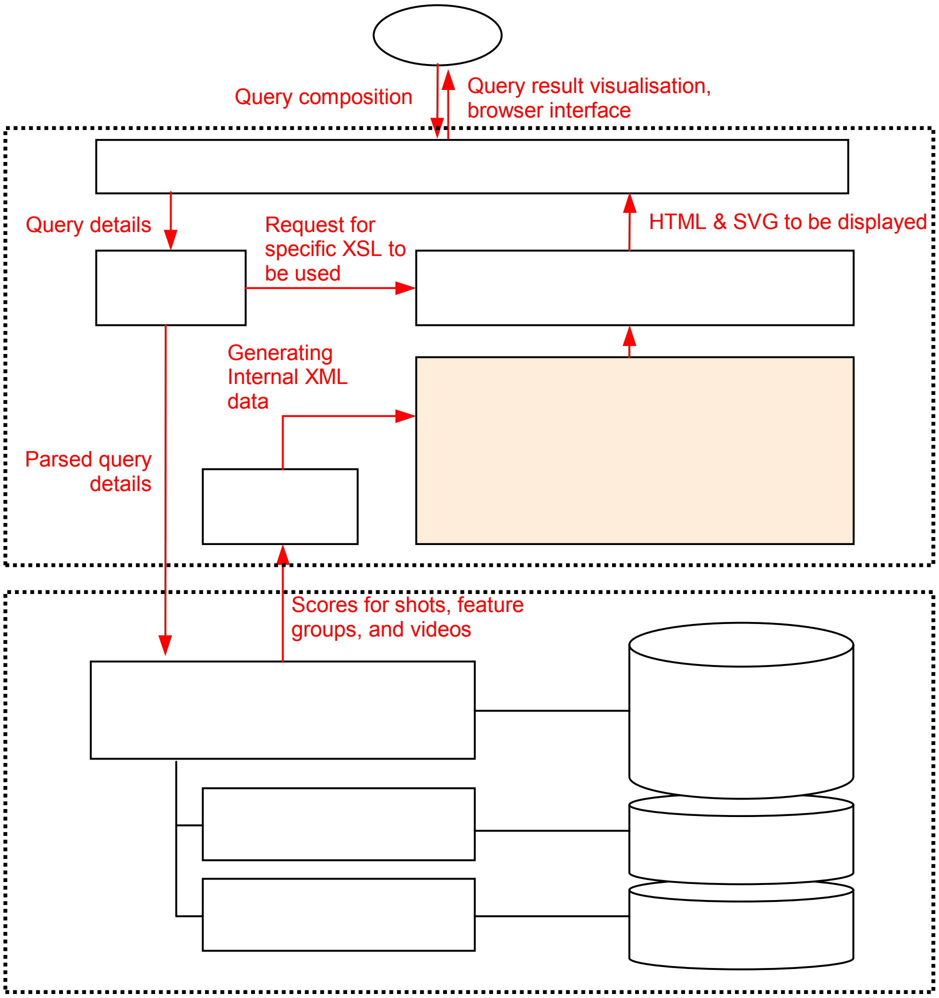

Query composition
Query result visualisation, browser interface
Query details
Request for specific XSL to be used
HTML & SVG to be displayed
Generating Internal XML data
Parsed query details
Scores for shots, feature groups, and videos

Figure 1: System architecture

3.1.2.1 Search and Retrieval of Video Units

Recall that the first phase of a user's search was to generate a ranked list of videos in response to the query. Each video is represented by an overall feature weight for each of the 10 features, which was generated by calculating the aggregate scores for each feature from each shot within that video and then dividing these aggregate scores by the total number of shots in the video.

Without having carried out a sampling of the accuracy of the feature detection we were using and given that our features originated from 3 separate participating groups (with large variations in average feature confidence) we normalised the weights of each feature so that no one feature would outweigh any other feature due to differences in confidence levels. In addition, we weighted each feature's influence based on its usefulness as an aid to distinguishing between different videos. For this we utilised a variation of the conventional text-ranking technique *idf*. This allowed us to increase the weighting of features that are better able to support distinguishing between relevant and non-relevant videos. In this way we weighted features that were better able to distinguish between videos higher than features that occurred in all or virtually all videos.

In response to a user's query, a ranked list of videos is returned to the user for further consideration. The overall rank for each video was based on linear combination of required (as specified in the query) feature influence along with the ASR search score. The influence of the ASR text in the video retrieval phase was weighted as 4 times higher than any of the other ten features – this being our best-guess parameter reflecting our belief that ASR transcript would be the primary method of ranking videos.

3.1.2.2 Search and Retrieval of Shot Units

Upon the user selecting a ranked video from the first phase for shot level examination, the query used to generate the ranked list of videos was augmented with an identifier of the chosen video and then sent to the search server in order to rank shots from within that particular video. The algorithm used to rank shots within a selected video is similar to that used to rank the videos with the following exceptions: the normalisation of feature weights for shots was calculated at a shot level as opposed to the video level; the weighting of each feature's influence was also calculated at the shot level and the ASR text scores were weighted at twice that of features in order to allow features to play a greater role in shot ranking than in video ranking.

When a user is examining a video at the shot level the ranking outlined above is only one of six sorting options available to the user. These six options discussed in 3.1.3 are chronological (using the ranking above for the SVG timeline as shown in Figure 2), combined (also using the above weighting but for shot ordering) and four feature groupings as discussed in 3.1.3 which do not use the shot level ranking described above.

3.1.3 Web Interface with XSL

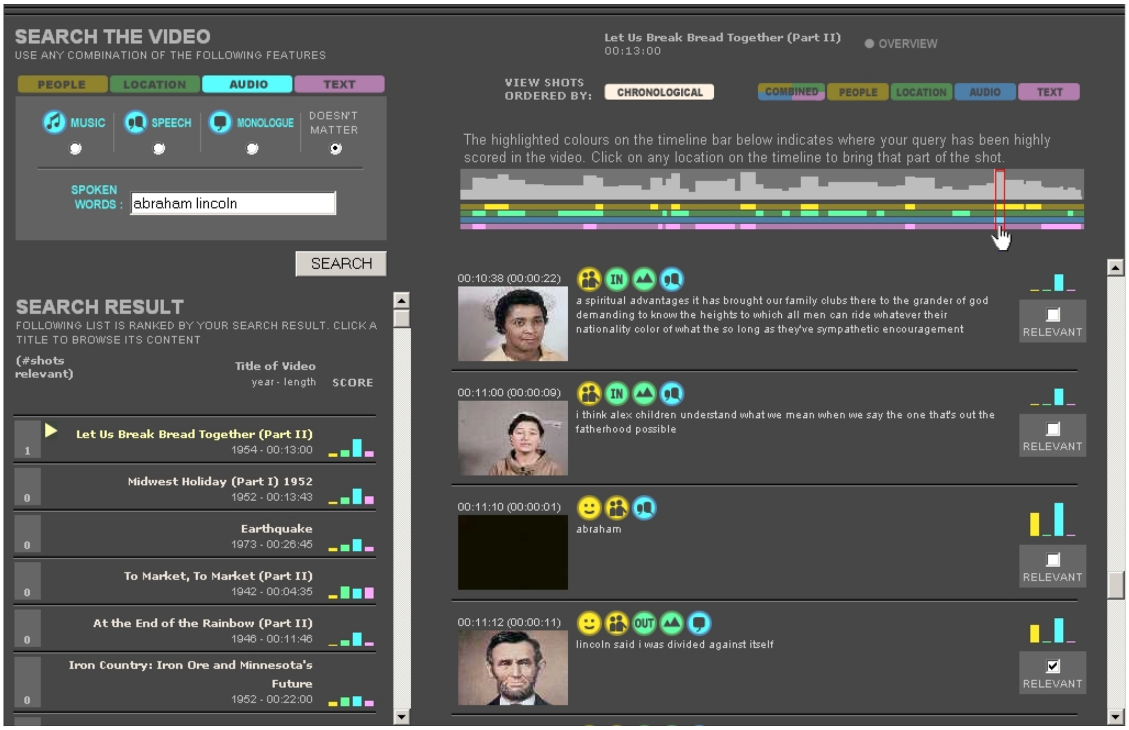
Having an internal XML-based architecture allowed us to clearly separate the presentation of data on the user interface from how the system works internally, significantly helping the system development process where software engineering and interface design can happen separately once the full XML format has been agreed.

For displaying on a web browser, XSL (eXtensible Stylesheet Language) has been extensively used on top of XML descriptions. XSLs were created when designing the interface (at design time), and used in conjunction with internal XML descriptions at user search time. XSLs transformed the internally generated XML video descriptions into 2 different formats based on a user's request – HTML and SVG (Scalable Vector Graphics). HTML is used to render most of the information display on the browser, including video listing with icons, score bars, ASR transcript, and other elements. This includes interactive elements such as ToolTips and JavaScript to enhance interaction. SVG is used to render a timeline on a chronologically displayed shot listing, plotting an indication of the matching status of the four feature groups against the user's query. Transforming to HTML means that any conventional web browser can be used to display the system's interface, though a SVG plug-in is required for viewing the SVG timeline and an Oracle plug-in is also required for streamed playback of video. Figure 2 shows a screen shot of the interface.

A user specifies her query on the query panel at the top left of the screen. All 10 features and ASR transcript query are grouped into 4 broad groups with distinctive colours associated with each. These are:

- **People**: Face(s), Group of People
- **Location:** Indoor, Outdoor, Cityscape, Landscape
- **Audio**: Music, Speech, Monologue, and ASR transcript search box
- **Text**: Text Overlay

Note that we included ASR transcript search as part of the third group. The query panel is organised by tabs, showing only one of the 4 feature groups at a time. In this way we expected to provide a simple and intuitive query screen to the users (4 features groups rather than 11 features) and the consequent retrieval result visualisation also makes use of the 4 grouping's colour schemes. The user specifies her query by clicking on the radio buttons for each feature, indicating if the feature is required or not. Clicking on the SEARCH button triggers retrieval (see section 3.1.2.1) and the result is displayed below the query panel, as a list of video programmes in a ranked order. For each video programme, score bars are presented indicating the relative scores of the 4 feature groups used in the user's query. Clicking on a title of the video in the list displays the content of the video on the right side and executes shot-level retrieval within that video. Initially an overview of the video programme is presented with the title, textual description and about 30 (temporally, not query focussed) selected keyframes. The user can further search for the wanted shots by clicking on the CHRONOLOGICAL button, which presents all of the chronologically ordered individual shots with the detected features, a keyframe, an ASR text portion, as well as score bars for the 4 feature groups. This is shown in Figure 2. Each of the shot entries also displays small round icons for the features detected for that shot, when their confidence value is above a threshold. At the top of the shot list in the chronological view, an SVG timeline is presented displaying the query matching status for each of the 4 feature groups as well as the combined score. The highlighted segment in the timeline indicates the part of the video that has matched against the query. The user can then click on the timeline to jump to the corresponding shot in the shot list below.



SPOKEN WORDS : abraham lincoln

(#shots relevant)	Title of Video year - length	SCORE
1	Let Us Break Bread Together (Part II) 1954 - 00:13:00	
0	Midwest Holiday (Part I) 1952 1952 - 00:13:43	
0	Earthquake 1973 - 00:26:45	
0	To Market, To Market (Part II) 1942 - 00:04:35	
0	At the End of the Rainbow (Part II) 1946 - 00:11:46	
0	Iron Country: Iron Ore and Minnesota's Future 1952 - 00:22:00	

The highlighted colours on the timeline bar below indicates where your query has been highly scored in the video. Click on any location on the timeline to bring that part of the shot.

00:10:38 (00:00:22)
a spiritual advantages it has brought our family clubs there to the grander of god demanding to know the heights to which all men can ride whatever their nationality color of what the so long as they've sympathetic encouragement
□ RELEVANT

00:11:00 (00:00:09)
i think alex children understand what we mean when we say the one that's out the fatherhood possible
□ RELEVANT

00:11:10 (00:00:01)
abraham
□ RELEVANT

00:11:12 (00:00:11)
lincoln said i was divided against itself
☑ RELEVANT

Figure 2: Web-based user interface.

The user can re-order the shots by combined score (see section 3.1.2.1), or by any of the 4 feature groups by clicking on the buttons beside CHRONOLOGICAL button, allowing quick access to shots in relation to a subpart of the query she specified. At any point while browsing, the user could click on a keyframe to start streamed playback of the video from that shot onwards, and this allowed the user to clarify if indeed a shot is relevant to a search topic. If a user finds a shot that she believes to be relevant to the search topic, she ticked a checkbox in each shot entry to indicate this, and the initial search result list (on the left of the screen) updated showing the number of shots she has indicated as relevant in the video programme.

3.2 Evaluation Procedure for Experiment

For our experiment we created a second version of our system, which supported only ASR transcript searching and not feature-based searching, in order to compare this to the full feature system. Our aim was to compare the two systems to see if the 10 semantic features helped users searching for a particular shot in the collection, as opposed to relying only on the text-based transcript searching. We also observed the interactive behaviour of users of the system in order to get other feedback from test users.

Twelve people participated as test users, 10 postgraduate students and 2 summer intern staff in the School of Computer Applications within the University. All had advanced levels of computer knowledge and familiarity with web-based searching, each conducting some form of online searching on a daily basis. Each of the 12 test users conducted all of the 25 query topics provided by NIST, one by one. The test users were divided into 2 groups, one group conducting the 25 topics in one order, and the other using the same topics but in reverse order. Six users used the full-feature system with all 10 features and the ASR transcript text searchable (3 forward and 3 in reverse order), and another 6 users used the system that had ASR transcript-only searching (also 3 forward and 3 in reverse order).

Each test user was seated in front of a desktop PC with headphones in a computer lab, and completed the first part of the questionnaire. We used the questionnaire developed over several years by the TREC Interactive track [10]. The questionnaire included pre-test questions, short post-topic questions, and post-test questions, which each of the users filled in at each stage of the testing. After a brief introduction, test users used a series of web pages which presented each topic, including the audio/image/video examples which form part of the topic descriptions. Users read, viewed, and played the examples that accompanied the topic and then conducted their search. Users were given 4 minutes for searching each topic and whenever a shot was located that the searcher thought answered the topic, they indicated this by checking the box beside the shot entry. At the end of the 4 minutes, users filled in a short post-topic questionnaire, and waited to be asked to start the next topic. The time taken to read the topic and examine the associated media elements was included in the four minute allocation per query. At the end of 12th topic, the users took 10-15 minute break with coffee/tea and light refreshments. After the break the next 13 topic searches continued, finishing with the post-test questionnaire. All individual users' interactions were logged by the system, and the results of users' searching were collected and from these results four runs were submitted to NIST for evaluation.

3.3 Submitted Runs

As mentioned above, we submitted four runs to NIST. These were the following:

1. *Full-feature system with all users (I_B_DCUTrec11B_1)*, where the selected shots of all users that used the full-feature system were aggregated (combined together) and this aggregated listing was sent as our first run to NIST.
2. *ASR transcript-only system with all users (I_B_DCUTrec11C_2)*, where the selected shots of all users who used the ASR transcript-only system were aggregated and the aggregated shot listing was submitted.
3. *User with highest number of shots selected in the Full-feature system (I_B_DCUTrec11B_3)*, where the results of the individual user who selected the highest number of shots using the full-feature system was submitted as our third run.
4. *User with highest number of shots selected in the ASR transcript-only system (I_B_DCUTrec11C_4)*, where the results of the individual user who selected the highest number of shots using the ASR transcript-only system were submitted as our fourth and final run.

3.4 Results of our Experiments

Figure 3 illustrates the average precision of each of our four runs. As can be seen the figures illustrate that no significant benefit in retrieval performance was found when the features were used in the search and retrieval process, which is the hypothesis we were testing. If we examine the user performance for the user with highest recall then it seems that the ASR+features interface aids the user more than the ASR–only interface, but with such a small number of users we can not say this with confidence.

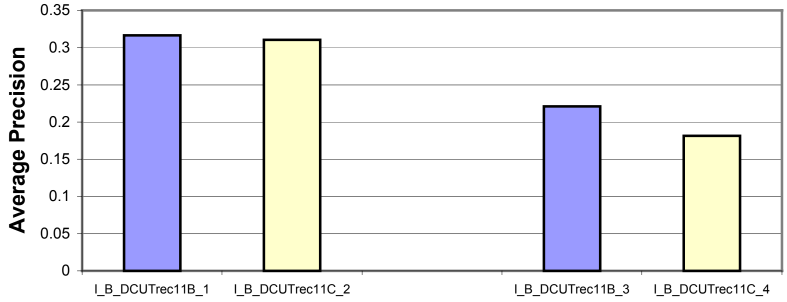


Figure 3. Average Precision of our four submitted runs

Post-experiment examination of the results of each user show us that while the feature interface worked well for some users, others had difficulties using it and the variance in recall attained by users of the feature interface was almost double that of the ASR-only interface.

Our observations after running the experiment suggest that a user's primary method of searching was using the ASR transcript, with features being used in addition when their inclusion seemed reasonable. This is clearly illustrated by examination of the two topics for which all 12 of our participants failed to find any relevant documents. Both of these topics (75 and 91) required search terms that were not in our chosen ASR transcript ('Rickenbacker' and 'parrot'), so when the ASR transcript could not aid retrieval, features were found to be of no significant benefit.

4. Conclusion

From the feature extraction task we observe that due to the nature of our approach to face detection, our system ran into difficulties operating on grayscale videos and slightly coloured material. Obvious improvements could be made by using a different approach which does not rely on skin colour segmentation. Our results for audio feature extraction showed our method worked very well and if we consider our full 1,000 identified shots instead of our 300 submitted ones, our performance was one of the best.

From the search task we find ourselves unable to come to any significant conclusions yet about the benefit of incorporating features into the retrieval process. More work needs to be done on methods of combining the features with the ASR transcript. In addition, the experiment has illustrated to us the need to provide users with query-focussed overviews as opposed to our overviews that used 30 temporally selected keyframes. Observations of the user experiments suggest that some users will not examine a video at the shot level if the overview does not show relevant keyframes regardless of the video's ranked position. Finally, our system seemed to operate very well as a browsing tool supporting search, however we do wonder whether a user needs to go through video level ranking before examining shots. Further experimentation into direct shot-based ranking across videos would answer whether this supports faster resource discovery or reduces the high variability of user performances we observed in our experiment.

Acknowledgements: We would like to thank the groups from IBM, Microsoft Research Asia and LIMSI who each provided some of the feature detection output that we used in our search submissions. The support of the Enterprise Ireland Informatics Research Initiative is also gratefully acknowledged.

References

[1] Jarina, R., Murphy, N., O'Connor, N., and Marlow, S., "Speech-music discrimination from MPEG-1 bitstream", in Kluev, V.V., and Mastorakis, N.E. (eds.). *Advances in signal processing, robotics and communications*, WSES Press, 2001, pp. 174-178.

[2] Jarina, R., O'Connor, N., and Marlow, S, "Rhythm Detection for Speech-Music Discrimination in MPEG Compressed Domain", *Proc. of the IEEE 14th International Conference on Digital Signal Processing DSP 2002*, Santorini, Greece, July 2002, pp. 129-132

[3] ISO/IEC JTC 1/SC 29/WG 11, "Information Technology — Multimedia Content Description Interface — Part 4", Audio, March 2002

[4] Yang, M.H. and Ahuja, N., "Detecting Human Faces in Color Images" *Proc. IEEE Int'l Conf. Image Processing*, October 1998, pp. 127-239.

[5] Yang, M.H., Kriegman, D.J., and Ahuja, N., "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(1), January 2002, pp. 34-58.

[6] Sobottka, K., and Pittas, I., "A novel method for automatic face segmentation, facial feature extraction and tracking ", *Signal Processing: Image Communication* **12**, 1998, pp.263-281

[7] Jain, A.K., "*Fundamentals of digital image processing*", Prentice-Hall, NJ, 1989.

[8] Lee, H. and Smeaton, A.F., "Designing the user interface for the Físchlár Digital Video Library". *Journal of Digital Information*, **2**(4), May 2002.

[9] Savoy, J., and Rasolofo, Y., "Report on the TREC-9 Experiment: Link-based Retrieval an Distributed Collections". *Proceedings of the 9th Annual TREC Conference*, November 16-19, 2000.

[10] TREC Interactive Track. Available online at URL: http://www-nlpir.nist.gov/projects/t11i/t11i.html
(last visited October 2002).

[11] Gauvan, J.L., Lamel, L., and Adda, G. "The LIMSI Broadcast News Transcription System", *Speech Communication*, 37(1-2):890198, May 2002.