

# TRECVID 2004 Experiments in Dublin City University

**Eddie Cooke, Paul Ferguson, Georgina Gaughan, Cathal Gurrin, Gareth J.F. Jones, Hervé Le Borgne, Hyowon Lee, Seán Marlow, Kieran Mc Donald, Mike McHugh, Noel Murphy, Noel E. O'Connor, Neil O'Hare, Sandra Rothwell, Alan F. Smeaton, Peter Wilkins**

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

## Abstract

*In this paper, we describe our experiments for TRECVID 2004 for the Search task. In the interactive search task, we developed two versions of a video search/browse system based on the Físchlár Digital Video System: one with text- and image-based searching (System A); the other with only image searching (System B). These two systems produced eight interactive runs. In addition we submitted ten fully automatic supplemental runs and two manual runs.*

### A.1, Submitted Runs:

- *DCUTREC13a\_{1,3,5,7}* for System A, four interactive runs based on text and image evidence.
- *DCUTREC13b\_{2,4,6,8}* for System B, also four interactive runs based on image evidence alone.
- *DCUTV2004\_9*, a manual run based on filtering faces from an underlying text search engine for certain queries.
- *DCUTV2004\_10*, a manual run based on manually generated queries processed automatically.
- *DCU\_AUTOLM{1,2,3,4,5,6,7}*, seven fully automatic runs based on discrete language models operating over ASR text transcripts and visual features.
- *DCUauto\_{01,02,03}*, three fully automatic runs based on exploring the benefits of multiple sources of text evidence and automatic query expansion.

*A.2, In the interactive experiment it was confirmed that text and image based retrieval outperforms an image-only system. In the fully automatic runs, DCUauto\_{01,02,03}, it was found that integrating ASR, CC and OCR text into the text ranking outperforms using ASR text alone. Furthermore, applying automatic query expansion to the initial results of ASR, CC, OCR text further increases performance (MAP), though not at high rank positions. For the discrete language model-based fully automatic runs, DCU\_AUTOLM{1,2,3,4,5,6,7}, we found that simple linear interpolation language models perform marginally better than other tested language models and that combining image and textual (ASR) evidence increases performance (MAP) over textual models alone. For our two manual runs we found that employing a face filter disimproved MAP when compared to employing textual evidence alone and that manually generated textual queries improved MAP over fully automatic runs, though the improvement was marginal.*

*A.3, Our conclusions from our fully automatic text based runs suggest that integrating ASR, CC and OCR text into the retrieval mechanism boost retrieval performance over ASR alone. In addition, a text-only Language Modelling approach such as DCU\_AUTOLM1 outperforms our best conventional text search system. From our interactive runs we conclude that textual evidence is an important lever for locating relevant content quickly, but that image evidence, if used by experienced users can aid retrieval performance.*

*A.4, We learned that incorporating multiple text sources improves over ASR alone and that an LM approach which integrates shot text, neighbouring shots and entire video contents provides even better retrieval performance. These findings will influence how we integrate textual evidence into future Video IR systems. It was also found that a system based on image evidence alone can perform reasonably and given good query images can aid retrieval performance.*

## 1 Introduction

This year the Centre for Digital Video Processing at Dublin City University participated in the TRECVID 2004 Search task only. We submitted eight interactive runs, ten fully automatic runs and two manual runs. For the interactive runs we developed an interactive web-based video retrieval system and compared the performance of two versions of the system in an interactive retrieval experiment. The system's underlying architecture is based on the Físchlár Digital Video System: we have used this backbone system to develop variations of interactive systems for long-term deployment experiments such as Físchlár-TV (Lee & Smeaton, 2002) and Físchlár-News (Smeaton *et al.* 2004), and also for a series of intensive short-term search experiments such as our previous Físchlár-TREC systems used for TRECVID Interactive experiments (Browne *et al.* 2001; Gaughan *et al.* 2003; Gurrin *et al.* 2004).

The Search test collection consisted of 64 hours of CNN Headline News and ABC World News Tonight broadcast recorded during the last half of 1998. Along with the video data in MPEG-1 file format, we have made use of the common Shot Boundary reference from CLIPS-IMAG, the Automatic Speech Recognition (ASR) transcript from LIMSI (Gauvain *et al.* 2002), Closed Caption (CC) transcripts from NIST and Optical Character Recognition (OCR), optical motion and face features all donated by CMU. The completed system was then used to conduct an interactive search experiment over a three day period using 16 experienced users.

The rest of the paper is organised as follows. In Section 2 we describe the developed interactive system’s architecture, retrieval and weighting mechanism, and user-interface design strategy. Section 3 then describes the experimental setup and procedure we took to conduct the Search task with the system, and initial results of our submission. Following on from that, section 4 describes our fully automatic and manual runs before concluding with a discussion of our findings from this years TRECVID participation.

## 2 Interactive Search Experiment

We developed two variations of the system to compare against each other. The first variation (System A) provides the users with text querying functionality and shot content-based relevance feedback, whereby the user starts by typing in some text query and/or adds in video/image examples that come with the topic, and during the search she can add any keyframe from the video into the query panel for subsequent querying. The second variation (System B) relied solely on relevance feedback based on keyframe images without any text-based querying. Thus the only way the user can conduct a search in the second system is to begin by adding in video/image examples into the query panel and trigger searching and subsequently add better keyframes into the query panel to improve the search result.

### 2.1 Interactive System Architecture

The interactive video retrieval system we developed for this year’s TRECVID has an XML-based architecture using MPEG-7 compliant video descriptions internally, and is somewhat similar to the TREC2003 system in terms of its major feature, that of relevance feedback using shot content. However, the visual features employed for calculating keyframe-keyframe similarity, along with changes in the underlying text engine has produced a system that is very different from TRECVID2003 and allows our user considerably more control over system operation.

Figure 1 shows the architecture of the interactive system, and will be explained throughout the following sections.

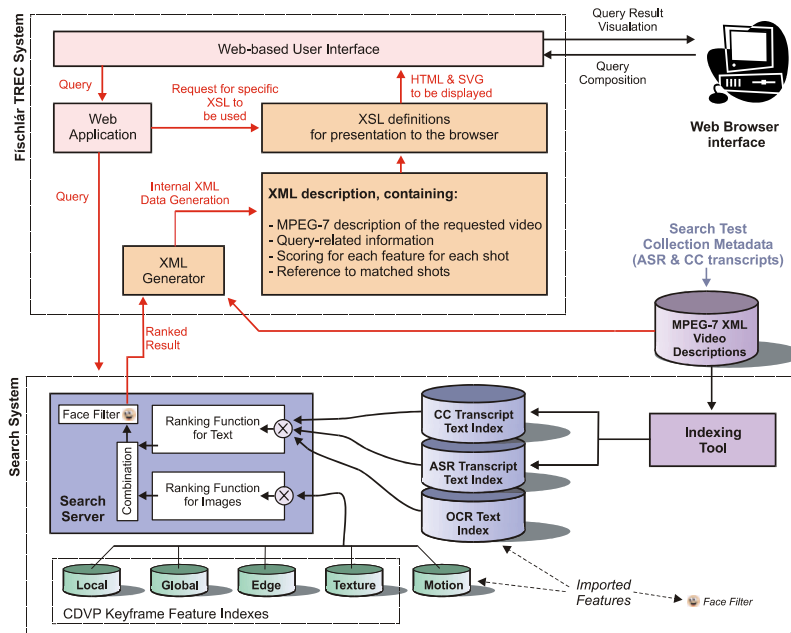


Figure 1: Físchlár-TREC2004 Architecture

### 2.2 Retrieval and Weighting Scheme

There were two essential aspects of the retrieval and weighting scheme for the system: the text search aspect and the image search aspect. Due to the nature of our experimental focus, both text and image ranking could be used singularly or in parallel. The same search engine was used for both systems A & B. In the image search system (System B), for example, the text engine would never be queried, even when the user selects interesting shots to feedback into the query. In addition (System A) a user may decide for certain queries to rely entirely on text and ignore image search altogether.

We will discuss text and image search facilities, combination of feature evidence and the feedback mechanism in the following sections.

#### 2.2.1 Text Search Facility

To support our text search facilities we used three sources of evidence, ASR, CC and OCR. For each source of evidence we implemented a separate search engine which operated using the BM25 ranking algorithm with appropriate weights (see bottom half of Figure 1). The underlying text search engine (Físréal) was developed within the CDVP and used to support many other

TREC experiments (Blott et al., 2004). All text indexed was processed to remove stopwords (SMART list) and stemmed using Porter's algorithm.

When processing a query, the system sends the query to each of the three search engines and combines the ranked output (top 500) of each engine to produce a final ranked list. The similarity scores of each engine were combined (addition) to produce a final document score. This final document score was subjected to a smoothing phase where the scores of the neighbouring documents were appended to the score of an individual ranked shot. The preceding two shots and the succeeding two shots added a fraction of their scores to the overall shot score with immediate neighbours adding 15% of their score and the second level neighbours adding 10%.

Finally, if in System A, a user wished to have the textual element of a query ignored (concentrating only on the visual aspects), then simply leaving the text field of the query interface blank will accomplish this as the search system will not process any text queries. The final outcome of the text search engine was a ranked list of 1,000 shots (with scores smoothed) generated by querying all three text search engines. These results can be used to rank shots before they are presented to the user, or alternatively (depending on the user requirements as expressed in the query) combined with the results of an image search phase to generate a final ranked shot list.

## 2.2.2 Image Search Facility

In order to support keyframe matching we developed an image search engine which processed all keyframes using the automatically extracted feature descriptors described below. The first four descriptors (below) were developed within the context of the aceToolbox, a toolbox of low-level audio-visual analysis tools being developed as part of DCU's participation in the EU aceMedia project (URL: <http://www.acemedia.org>).

- *Local Colour Descriptor* (Colour Layout - CLD) is a compact and resolution-invariant representation of colour in an image. The feature extraction process consists of four parts; first, the image is partitioned in 64 (8x8) blocks; second, the representative colour of each block is determined by using the average colour in each block; third, a DCT transform is applied to these three (one for each of the colour components) tiny image icons of size 8x8, resulting in three sets of 64 coefficients; last a few low-frequency coefficients are selected using zigzag-scanning and nonlinearly quantized to form the CLD. In practice, 6 coefficients are retained for luminance and three for each chrominance.
- *Global Colour Descriptor* (Scalable Colour - SCD) measures colour distribution over an entire image. It is defined in the hue-saturation-value (HSV) colour space and produces a 256-bin colour histogram, normalised, non-linearly mapped into a four-bit integer value, and then encoded by a Haar transform. This last consists of computing the sum and the difference of adjacent pairs. The sum of adjacent bins leads to a histogram with half the number of bins. Repeating this process four times, we finally obtain a 32-bin histogram. Another form of scalability is achieved by scaling the quantized representation of the coefficients to different numbers of bits. Here the three less significant bits were discarded.
- *Edge Histogram Descriptor* (EHD) is designed to capture the spatial distribution of edges by dividing the image into 4x4 subimages (16 non-overlapping blocks) and then edges are categorized into 5 types (0°, 45°, 90°, 135° and "nondirectional") in each block. The output is a 5-bin histogram for each block, giving a total of 5x16 = 80 histogram bins.
- *Homogenous Texture Descriptor* (HDT) describes directionality, coarseness, and regularity of patterns in images. It is computed by first filtering the image with a bank of orientation and scale sensitive (Gabor) filters, and then computing the mean and standard deviation of the filtered outputs in the frequency domain. In this work we only use the mean values to compute the similarity between the images.
- Motion – based on CMU donated motion feature.
- Face Filter, based on CMU donated feature, but is employed as a shot filter and is discussed separately in Section 2.2.4.

More details on the first four descriptors can be found in (Manjunath *et al.* 2001). The similarity between images was estimated by the  $L_2$  Minkowsky (Euclidean) distance for each of the features. In fact, since all the outputs of the descriptors are quantized, they are integer values. In this case, the  $L_1$  distance would conduct to many equal similarities between images, while the  $L_2$  one results in a larger range of distances, that benefits the indexing process.

Five features (excluding face filter) were the underlying methods by which we could compute keyframe to keyframe similarity, whether these keyframes originate from within the video collection or come from the topics. In an interactive environment a user could add images from the topic description to the query for processing by the image engine. In addition, by employing feedback, a user may also add shots (represented by keyframe images) from within the collection to the query. In this way a user query may consist of any number of images either from the topic description or from the collection itself.

At query time, the user could select which (or all) of the five features were important via the interface (see Section 2.3.2) and each of these features were combined together to produce a final feature ranked list. It is hoped that analysis of the user logs of the experiment should illustrate which features users employed when searching different topics. If one or more features were selected by the user, the query images were used to identify similar shots by generating separate ranked lists of the top 500 shots for each feature, for each query image. These ranked lists were combined into a single ranked list by examining shot rank position within the ranked lists as opposed to the actual similarity scores. This was done so as to avoid any combination issues due to very dissimilar score distributions among the required features.

The overall outcome of the Image search element was a final ranked list of the top ranked 1,000 shots from all features selected for the query images. See Section 2.2.5 for a description of the combination mechanism for image evidence with text evidence.

### 2.2.3 Relevance Feedback Mechanism

A central aspect of the system is that the users could select shots to be appended to the query in a form of relevance feedback. For example, if a user queries for 'ice hockey' using a text only query and locates a number of good examples of shots of ice hockey playing, then one or more of these shots can be added to the query and included in subsequent searches. Feedback works in different ways for text and for images. For text, any shot that is fed into the query contains candidate words that can be used to generate an extended query. These candidate words are the ASR and the CC text transcripts of that shot from which we removed stopwords and stemmed (SMART and Porter again). If a number of shots are feedback into the system the candidate words are the combination of the ASR and CC text from all these shots.

The top ranked 10 words from these candidates are then chosen using RSV selection and appended to the original query, with the original query terms weighted three times higher than the appended terms. This new text query is then processed in the normal manner (see Section 2.2.1).

For image feedback, the process is different in that the keyframe of the associated shot is simply used as another keyframe for the image matching engine to process (see Section 2.2.2). For example, if the original query contained a single sample keyframe from the topic description, and a user appended three additional shots to the feedback query, then the query would simply consist of four sample images, each processed in a similar manner and weighted equally.

### 2.2.4 Face Filter

In addition to the search options described above, we also included a face filter which the user could employ as required, and was presented in the interface in the same manner as the other image similarity features. This filter, based on the donated feature by CMU (Schneiderman 2000), filters the final ranked output of the search engine to remove shots that do not contain faces.

A preliminary examination of the performance of this filter can be found in our manual run (**DCUTV13\_9**) in which we applied a face filter to a normal three-engine text search result. The addition of the face filter was manually chosen for each query (hence a manual run as opposed to a fully automatic). The results of this filter illustrated a surprising decrease in performance of the system when face filter is applied. MAP drops from 0.060 (**DCU\_auto03**) to 0.038 with Recall dropping from 448 to 299. Perhaps this is a by-product of employing text smoothing, but we need to explore this further.

### 2.2.5 Combining Image and Content Evidence

Depending on the user's query or system used, image and text evidence (ranked lists) may need to be combined together to generate an overall ranked list for the system to present to the user. This combination was done by rank position of shots from within both lists and the top 200 shots were returned for presentation to the user. In situations where text was not required (e.g. System B) the ranked list was simply the top 200 shots from the image engine, and similarly for the text engine alone.

## 2.3 User-Interface Design

The design of the user-interface started early in the system development process, and involved several phases of iterative refinement each of which consisted of a series of screen mock-ups and discussion within the group, after which the mock-ups were refined based on the discussion. While the main search feature of the system (adding a shot content into the Query Panel for subsequent querying) was similar to our previous year's system (Gurrin *et al.* 2004), from the iterative refinement process we tried to enhance what we considered as strength from our interface from last year, and at the same time to rectify the problem elements we identified from last year's user experiment.

In this section, we describe the overall interaction scheme adopted, the features integrated to help the user search the videos, and the strengthened browsing facilities incorporated into the system.

### 2.3.1 Overall Interaction Scheme

Because of the nature of the target task (searching for video clips for a limited period of time as specified in the topic description), there needed to be task administrative elements (task number, clock showing remaining time, task description, and list of relevant shots saved by the user) and actual search/browse tools (query panel, search result display, playback, etc.). Thus we started by grouping the necessary elements on the screen into two: Administrative Area and Work Area.



Figure 2: Overall interface: browsing initial search result (System A)

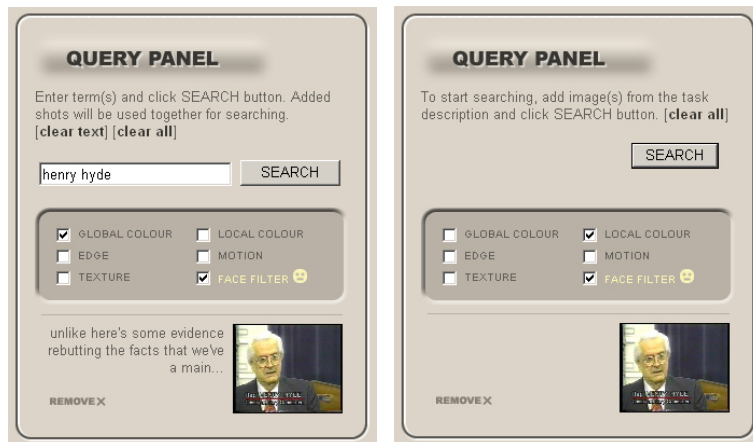
Figure 2 shows the overall interface of System A (ASR & image): the background is dark grey, on which a large work panel in light brown colour sits and occupies most of the screen real estate. This panel is the Work Area where the user will be working on searching/browsing of videos; the dark grey area on the right side of the screen is the Administrative Area and left for displaying most of the administrative elements. The Work Area is again divided into a **QUERY PANEL** area and **SEARCH RESULT** area, separated by bevelled lines and prominent labels. The look-and-feel of the interface is characterised by 3-D buttons that imply interaction possibility, round edges, panel protrusion, and bevelled lines that separate the areas; all these elements have similar visual characteristics that combine all the elements on the screen into a coherent unified interface; they are there not to distract the users or merely to look “cool”, but rather to organise the areas and give a unique, coherent and consistent theme which adds to the user experience and satisfaction.

Most of the Administrative Area is occupied by the list of saved shots: as the user finds shots that she thinks satisfy the topic, she adds them into the list of saved shots. As she finds more and more of the shots and adds them, the list will grow. Because this saved shot list is visible at all times, she can easily relate the shots found in the Work Area during searching/browsing and compare, add to Query Panel or delete. The Administrative Area also displays a miniature clock that shows the remaining time for each topic task, and a button to re-display the topic description.

### 2.3.2 Querying with Relevance Feedback

At the beginning of each topic search, the user is presented with the topic description with the example videos/images on the screen. The user may play the example videos to understand the topic more clearly, which the interface encourages by emphasised play buttons above each example video (not shown in Figure 1).

In the case of System A (ASR & image) system, after the user understood the topic, she starts typing in some keywords in the text box, and adds in the example videos/images into the Query Panel (see



(a) Text query and example video/image or any keyframe added in for subsequent search (System A)

(b) Image-only querying - the only way to query is by adding images (System B)


Figure 3: Query Panel

Figure 3(a)). In the case of System B (image-only), there is no text box (see Figure 3(b)). When the example videos/images are added, the user is required to check at least one of the checkboxes just below text box which contains 6 checkboxes, corresponding to the 6 visual features used for the visual retrieval process described in Section 2.2.2.

Once a query is composed, the user clicks on the **SEARCH** button, which will trigger retrieval and the result will be displayed on the **SEARCH RESULT** area.

### 2.3.3 Examining the Search Result

The search result is displayed in the middle column of the screen (Figure 2), with each entry composed of 5 shots (Group of Shots) where the middle one is the matched shot, and neighbouring shots (two preceding and two following) presented to provide the context of the matched shot. The matched shot's keyframe is displayed largest and the neighbouring 2 keyframes on both sides smaller, to give the idea of matching emphasis to the middle. This was one of the refinements from last year's system which showed all 5 keyframes in same size that sometimes caused confusion to the users. As Figure 2 shows, the matched shot's keyframe (the middle one among the five keyframes) also has a red surrounding box to highlight its matched status, while the matched text term(s) in the ASR transcript is also highlighted in red below the keyframes.

Each entry displays the name and date of the broadcast, and also the approximate location of the matched point within the broadcast as a timeline (  ). In this timeline, the red line's horizontal position indicates the relative location of the match within the broadcast, while the line's height indicates the match score or confidence. This bar was also newly introduced this year to help the user orientate where the matching part is within a broadcast and also provide a clue to the confidence of the matching, so that the user can decide whether to look more into this broadcast or not. In the future versions, we are considering also to indicate in this timeline *all other matches within this broadcast* with multiple red lines - this will further help the user decide whether to browse the broadcast further or not.


### 2.3.4 Searching to Browsing

Nearby a matching shot or somewhere within the broadcast, there is a likelihood that there will be more relevant shots. By providing a mechanism to see the full broadcast from which the matched shot was found, the system can allow more efficient browsing. However, going into a full broadcast (most of them about 25-30 minutes long) is at the same time making the user's browsing space considerably larger, thus will require more user effort. We have observed from last year's experiment that this was indeed the case and thus found the users seldom browsed at the full broadcast level, even though there could have been a good chance to find more relevant shots in it. From this observation, this year we added an intermediate level (between search result browsing and full broadcast browsing) where all matched shots within a broadcast is selected and displayed. From an initial search result, the user looks at one matched entry, and clicks on "**>>MORE MATCHES IN THIS BROADCAST**" (just beside the timeline in Figure 2) to see all other matches within this broadcast, replacing the initial search result.

Without having to go into the full broadcast level browsing, the user can find more matched points conveniently presented by the system. After browsing this screen if the user thinks there could be further more matches within other parts of this broadcast, she clicks on "**>> BROWSE FULL BROADCAST**" link from this level (not shown in Figure 2), which will bring her into the full broadcast browsing. In full broadcast browsing, an interactive SVG (Scalable Vector Graphics) timeline is presented with the matched points in the broadcast highlighted, to allow the user to quickly jump within the broadcast.

Thus, the interface provides 3 levels of browsing: initial search result, more matches within a broadcast where a match occurred, and full broadcast - this is to support the user more efficiently browse the wanted shots.

### 2.3.5 Refining the Query with Relevance Feedback

At all levels of browsing, there is a pair of buttons (  ) below each keyframe as Figures 2 shows. The grey button on the left is **ADD TO QUERY** button: at any point in the browsing, the user can click this button to use this keyframe as part of the next query. As explained in Section 2.3.2, the added shot is listed in the Query Panel, and will be used when the user clicks on the **SEARCH** button again. The button on the right is the **SAVE** button: at any point in the browsing, if the user thinks this shot answers the topic task, she clicks on this button to add this shot into the saved shot list on the Administrative Area, on the right side of the screen. The colour of the button (dark grey background and yellow text) is designed to match that of the Administrative Area where the saved shots will be listed. On the saved shot list, for each of the saved shots there is a grey **ADD TO QUERY** button - at any point in the browsing, the user can click this button to add this shot into the Query Panel. By having the **ADD TO QUERY** button on the saved shot list, the user can more readily add/remove the added shots to/from the Query Panel, experimenting the search result very quickly.

## 2.4 Experiment

We recruited sixteen test users for our experiments from within the Centre for Digital Video Processing, excluding the system developers from our selection. All of our participants therefore had high levels of computer experience and varying levels of experience within the field of Information Retrieval. All would have been expected to understand how to use image and text search engines such as those implemented for the interactive experiment.

A number of days prior to the start of the experiments, participants were granted remote and unsupervised access to our search system. We encouraged them to familiarise themselves (for an hour or so) with both system variants and their features, by searching for topics of their own interest, or making use of the four sample topics that we formulated for training purposes. In addition, before starting the actual experiment, each user was again given another opportunity to complete training searches for the four sample topics (two per system variant) under supervised experimental conditions, with a system developer on hand to answer any questions. Thus, we can classify our participants as "experienced" users, compared to last year's experiments where



users were computer science postgraduate students, though not necessarily familiar with video retrieval. Also, last year, they received only a 10-minute tutorial on how to use the systems and completed just two sample queries before beginning the search experiment. By carrying out the experiments using more expert users, we reduced the problems caused by the fact that the performance of a system is determined by the capabilities and efficiency of its users.

This year, slightly under 15 minutes was allocated for the completion of each task (topic), including the training tasks, compared to the 7 minutes of last year. This allocated time included the time taken to read the TRECVID topics. This increase impacted on the way we executed our search sessions. As each session was going to be significantly longer, we decided to allow users to take breaks in-between search tasks, should they feel fatigued, and allowed them to split their session in two and spread it over several days, if they so wished.

## 2.4.1 Experimental Design Methodology

Similarly to last year, this year’s guidelines for the interactive Search Task included guidelines for the design of user experiments (TRECVID guidelines, 2004). These guidelines were developed in order to minimise the effect of user variability and possible noise in the experimental procedure. The guidelines outlined the experimental process to be followed when measuring and comparing the effectiveness of two system variants (System A and System B) using 24 topics and either 8, 16 or 24 users, each of whom searched 12 topics. A user does not see the same topic more than once and a user completes all work on one system variant before beginning any work on the other variant. We followed the TRECVID guidelines for evaluating 16 users searching 12 topics using 2 system variants. Users were assigned randomly and the order of topic presentation for a given system and user was also randomised. This design allows the estimation of the difference in performance between two system variants run at one site, free and clear of the main (additive) effects of user and topic and the gathering of some information about interactions.

## 2.4.2 Experimental Procedure

Each user was seated in front of a desktop PC in a computer lab. Before beginning the experiments, each user completed the standard pre-search questionnaire as required for interactive TRECVID experiments. The questionnaire included pre-search questions, short post-topic questions and post-search questions, which each of the users filled in at each stage of the session. After a logging on and conducting test search on the four training topics, the users began the topic search. Each user was presented on the system interface with the topic description and the video/image examples. Users view and/or read, and were free to play the examples that accompanied the topic and then conducted their search.

As stated the users were given just under 15 minutes for each topic, and when a shot a user thought answered the topic was located, they indicated this by clicking the **SAVE** button underneath the shot’s keyframe as explained in Section 2.3.5. At the end of the allocated time, the users filled in the post-topic questionnaire, and started the next topic, provided they did not want to take a break to rest or get refreshments. After a user completed their 12 topics, they filled in the post-test questionnaire. All individual users’ interactions were logged by the system, and the results of users’ searching (saved shots in the Administrative Area) were collected and from these results eight runs were submitted to NIST for evaluation.

## 2.4.3 Submitted Runs

For the interactive search task, we submitted eight official runs for evaluation. These runs were based on the results of our 16 users, where each user processed 12 of the 24 topics. Therefore our runs used the output of 4 different users (6 topics each) on a single system resulting in four runs for both System A & B. Our runs were labelled **DCUTREC13a\_{1,3,5,7}** for System A and **DCUTREC13b\_{2,4,6,8}** for System B. Users were combined to generate runs sequentially without any intentional selection of users to improve retrieval performance.

## 2.4.4 Results of the Interactive Experiments

System A outperformed System B and this was to be expected. The average MAP over all four runs for System A was over twice the average MAP of System B.

**Table 1:** Results of the Interactive Runs

Run	System A (text & image)				Run	System B (image only)			
	MAP	MP@rel	P@10	Recall		MAP	MP@rel	P@10	Recall
DCUTREC13a_1	0.203	0.532	0.683	0.222	DCUTREC13b_2	0.066	0.182	0.506	0.082
DCUTREC13a_3	0.190	0.481	0.665	0.181	DCUTREC13b_4	0.094	0.250	0.504	0.097
DCUTREC13a_5	0.191	0.506	0.652	0.180	DCUTREC13b_6	0.088	0.215	0.395	0.083
DCUTREC13a_7	0.133	0.397	0.564	0.139	DCUTREC13b_8	0.074	0.195	0.314	0.077
AVG	0.179	0.479	0.641	0.181	AVG	0.081	0.211	0.430	0.085

However examining MAP for both systems, the average deviation about the mean for System A (at 0.023) is over twice that of System B (0.011). What this suggests is that System A brings our variances in user ability more than System B, although we can not be sure if this is because System B finds less relevant shots. Were we to examine MAP distribution as a function of the number of relevant shots found by B compared to A then the differences between the average deviations disappear.

Average recall for A and B clearly illustrate that not using text reduces recall of B to 47% of A (taking the median as opposed to the average recall gives a value of 46% which suggests that outliers do not affect this).

Examining the results on a per-topic basis using averaged MAP over all four runs for each system shows that for twenty of the topics (85%) System A performed better, while for three topics System B performed better. The topics for which System B

outperformed System A were topic 140 (4%), 142 (59%) and 144 (76%). These topics were looking for shots of bicycles, tennis players and Bill Clinton in front of the US flag. The subject of these topics (especially 142 and 144) could be considered to be visually striking and perhaps well suited to image based retrieval.

For seven topics System A notably outperformed System B (where System B’s average MAP was less than 25% that of System A). These topics were 126, 128, 133, 134, 137, 138 and 143. Four of these topics were looking for shots of people, without any distinguishing features (such as the US flag required).

### 3 Fully Automatic and Manual Search Experiments

We submitted ten fully automatic supplemental runs, three of which were runs based purely on our underlying text search engines. We were interested to see the effect of incorporating extra textual evidence and query expansion on top of a baseline ASR-only retrieval system. For these runs, the entire query was submitted unmodified to the search engines. Analysis of our text search engine performance illustrates that a search system that utilises all three sources of textual evidence (**DCU\_auto02**) outperforms a system based only on the ASR text (**DCU\_auto01**) by 17% and locates 54 additional relevant shots. A further fully automatic run (**DCU\_auto03**) suggested that implementing automatic query expansion using RSV to select 10 terms to be added to the query from the top ranked 10 shots, before processing the query, further increases average precision by 30% above the baseline **DCU\_auto01**. We did not implement this automatic RSV query expansion in our interactive system. These figures are presented in Table 1 below.

*Table 2: Comparing fully automatic text-only runs*

	MAP	Recall
DCU_auto01 (baseline)	0.046	298
DCU_auto02	0.054	352
DCU_auto03	0.060	448

Future work will be to examine the text engine used for our interactive experiments in light of our findings from these runs and other fully automatic runs (as outlined in Section 3).

In addition to these, we submitted 7 other fully automatic runs that use a discrete language modelling approach for both text- and visual-based video information retrieval. These are discussed in section 3.1 below.

In addition to these ten fully automatic runs, we submitted two manual runs:

- **DCUTV13\_9** to evaluate the use of the face filter (see section 2.2.4), and
- **DCUTV13\_10** to evaluate the benefit of manually generated queries (by removing unnecessary topic terms) over and above an equivalent fully automatic run, DCU\_auto03.

Based on the result of **DCUTV13\_10**, our findings naturally suggest that removing unnecessary topic terms increased average precision, though only marginally from 0.060 to 0.063, with recall remaining constant. Notable improvements in precision can be found at high rank positions. In an interactive experiment we would expect experienced users to be able to generate good queries and achieve these minor performance improvements.

#### 3.1 Fully automatic Language Modelling Experiments

Our language modelling approach is similar to that outlined by Westerweld *et al.* (2003), but instead of modelling our visual features using a multivariate Gaussian mixture model, we use smoothed multi-dimensional histogram representations for our three visual features HSV colour, Canny edge and DCT texture. The benefit of discrete feature representations in comparison to continuous representations for the language modelling approach to visual retrieval is the potential for a speedier calculation of the query-likelihood ranking function.

We compared three language modelling smoothing methods, simple linear interpolation, absolute interpolation and Dirichlet smoothing, for the text-based retrieval of video shots using the hierarchical video structure shot+adj+video – shot, adjacent shots (window of 9 shots) and video structure. Our text representation and the simple interpolation language model are essentially the same as Westerweld *et al.* (2003) but with different parameter settings. The TRECVID documents and topics were stopped using the SMART stopword list and stemmed with the Porter stemmer. For the TRECVID topics we supplemented the SMART stopword list with the following functional TRECVID topic words: “*find, additional, shots, scenes, pictures, containing, including, showing, lots, groups, multiple, partly, partially, visible*”. The three text-only runs produced very similar results. The simple linear interpolation language model (**DCU\_AUTOLM1**) achieved the best result with a MAP of 0.069, followed by the Dirichlet language model (**DCU\_AUTOLM3**) with a MAP of 0.066, and absolute interpolation language model (**DCU\_AUTOLM2**) with a slightly worse MAP of 0.065. We will now describe the technical details of these 3 runs.

The **DCU\_AUTOLM1 “text\_interpLM”** run is an ASR-only run using a simple linear interpolation language model for the shot+adj+video structure in which the probability of a word is given by

$$Pr_{interp}(w|\dots) = \lambda_{shot} * Pr(w|shot) + \lambda_{adj} * Pr(w|adj) + \lambda_{video} * Pr(w|video) + \lambda_{col} * Pr(w|col).$$

The mixture weights  $\lambda_{shot}=0.3$ ,  $\lambda_{adj}=0.1$ ,  $\lambda_{video}=0.1$  and  $\lambda_{col}=0.5$  were chosen by optimising the model on the TRECVID 2003 collection and topics.

The **DCU\_AUTOLM2 “text\_absinterpLM”** run is an ASR-only run using a language model for the shot+adj+video structure that is smoothed at each level with the absolute interpolation smoothing method. Absolute interpolation is defined as:

$$Pr_{abs}(w; \delta, doc, P_{bg}) = (r - \delta)/N + (((B - N_0) * \delta)/N) * P_{bg}(w) \quad \text{if } r > 0,$$



$$((B-N_0) * \delta)/N * P_{bg}(w) \quad otherwise,$$

where parameter  $\delta$  controls the amount of smoothing,  $P_{bg}$  is the background probability distribution,  $B$  is the size of the indexing language,  $doc$  is the given text document of size  $N$  in which the word  $w$  occurs  $r$  times and  $N_0$  is the number of zero frequency terms within the document. We hierarchically combine the shot+adj+video structure using this smoothing model: first the video structure is smoothed with the collection model; which is then used to smooth the *adj* structure; which is finally used to smooth the *shot* structure. This leads to the following definition for the probability of a word using this hierarchical absolute smoothed language model:

$$Pr_{hier\_abs}(w; \dots) = Pr_{abs}(w; \delta_{shot}, shot, Pr_{abs}(w; \delta_{adj}, adj, Pr_{abs}(w; \delta_{video}, video, Pr_{ML}(w|col))).$$

We set  $\delta_{shot}=0.3$ ,  $\delta_{adj}=0.8$  and  $\delta_{video}=0.95$ , which were optimised using the TRECVID 2003 collection.

The **DCU\_AUTOLM3 “text\_dirinterpLM”** run is an ASR-only run using a hierarchical Dirichlet smoothed language model for shot+adj+video structure. Dirichlet smoothing is defined as

$$Pr_{dir}(w; doc, \mu, P_{bg}) = (N / (N + \mu)) * P_{ML}(w|doc) + (\mu / (N + \mu)) * P_{bg}(w),$$

where  $\mu$  controls the amount of smoothing (number of virtual terms from the background collection), and  $doc$ ,  $N$ , and  $P_{bg}$  are as defined previously. As with absolute smoothing we apply this smoothing hierarchically on the shot+adj+video structure producing the following definition for the probability of a word:

$$Pr_{hier\_dir}(w; \dots) = Pr_{dir}(w; shot, \mu_{adj}, Pr_{dir}(w; adj, \mu_{video}, Pr_{dir}(w; video, \mu_{col}, Pr_{col}))$$

The parameters were optimised using the TRECVID 2003 collection, which produced the following values:  $\mu_{adj}=100$ ,  $\mu_{video}=2000$ ,  $\mu_{col}=20000$ .

We submitted two *visual-only* discrete language model runs based on simple linear interpolation and Lidstone smoothing. For the visual features we use languages consisting of all the discrete symbols in the features multidimensional histogram representation. Our visual features consist of Canny edges, HSV colour and DCT texture for 5x5 regions for all keyframes including those for sub-shots. We represent the Canny edge direction using 64 direction terms and an extra term for the non-edge pixels. We represent HSV colour using a 16x4x4 histogram and DCT texture using a 3x3x3x3x3 histogram for the first 5 DCT coefficients of brightness band. The simple linear interpolation smoothed language model (**DCU\_AUTOLM4**) achieved a MAP of 0.018, whereas the Lidstone smoothed language model (**DCU\_AUTOLM5**) achieved a marginally lower MAP of 0.017. We will now describe these runs in more detail.

The **DCU\_AUTOLM4 “visual\_interpLM”** is a visual-only run using simple linear interpolation smoothed language model for the 5x5 regional colour, edge, and texture visual features. The smoothing parameters for the separate language models for each feature were  $\lambda_{colour}=0.05$ ,  $\lambda_{edge}=0.55$ , and  $\lambda_{texture}=0.85$  which were chosen based on TRECVID 2003 collection. The top 500 results for each feature were given a normalised rank score from 1.0 to 0.0 based on their rank position in each of the features’ results:

$$score_{...} = \begin{cases} 1.0 - ((rank_{...}-1)/500) & \text{if rank} < 500, \\ 0 & \text{otherwise} \end{cases}$$

These scores from the three visual features were combined using simple linear interpolation (i.e. weighted Borda fuse):

$$score_{visual\_example} = w_{colour} * score_{colour\_example} + w_{edge} * score_{edge\_example} + w_{texture} * score_{texture\_example}$$

We used the following weights:  $w_{colour}=0.50$ ,  $w_{edge}=0.05$  and  $w_{texture}=0.45$ . The results from multiple visual examples for a topic were combined by averaging the rank positions of the top 500 results from each visual example (i.e. Borda fuse). When shots had multiple keyframes its best keyframe score was used.

The **DCU\_AUTOLM5 “visual\_lidstoneLM”** run is a visual-only run using Lidstone smoothed language model for the visual features. Lidstone smoothing is defined as follows

$$Pr_{lid}(w; doc, \lambda) = (r + \lambda)/(N + B\lambda),$$

where  $\lambda$  controls the amount of smoothing,  $r$  is the number of occurrences of the term  $w$  in the document,  $N$  is the size of the document and  $B$  is the size of the language. Lidstone smoothing adds a virtual count  $\lambda$  uniformly to each term and can also be viewed as a simple linear interpolation with a uniform source since the documents (keyframes) are a fixed size. The smoothing parameters for the separate language models for each feature were  $\lambda_{colour}=1.55$ ,  $\lambda_{edge}=1.55$  and  $\lambda_{texture}=0.07$ , which were chosen based on the TRECVID 2003 collection. We combine the scores for multiple features, examples and keyframes using the same procedure and settings as for the DCU\_AUTOLM4 run.

We submitted two *combined text and visual* runs based upon fusing the simple linear interpolation text language model with the simple linear interpolation visual language model in run **DCU\_AUTOLM6 “text\_interpLM + visual\_interpLM”** and with the Lidstone visual language model in run **DCU\_AUTOLM7 “text\_interpLM + visual\_lidstoneLM”**. For both runs the normalised rank positions of the top 1000 text results were combined with the normalised rank positions of the top 500 visual results (weighted Borda fuse).

$$score_{text\_visual} = w_{text} * score_{text} + w_{visual} * score_{visual}$$

We used the following weights:  $w_{text}=0.80$  and  $w_{visual}=0.20$ . The DCU\_AUTOLM6 run achieved a MAP of 0.078, which was overall the highest MAP for the TRECVID 2004 fully automatic search task and the DCU\_AUTOLM7 run achieved a marginally lower MAP of 0.077.

Overall, we found that the discrete language modelling approach achieves relatively effective performance in comparison with the results of other TRECVID participants for the fully automatic search task. We also observed that the simple linear interpolation language model has a marginal advantage over the other tested language models but these differences were not

significant for the visual-only features. We believe that these video retrieval results can be further improved by better visual features and fusion methods, which we are currently investigating.

## 4 Conclusions

Following on from our experimental runs in the fully automatic and manual search experiments we have identified scope to improve the quality of our text search engine for future interactive experiments. For example, optimising BM25 parameters, or incorporating an automatic relevance feedback mechanism (see Section 3) may help to improve performance. Our interactive experiments suggest that text IR still is the key feature for our interactive video retrieval system, but that image search can not only aid retrieval, but surpass text IR for some topics. Our image search system does seem to perform well (especially so for topics 142 and 144), but we will be improving this functionality before TRECVID 2005.

Our initial experiments into discrete language modelling for our fully-automatic runs found that simple linear interpolation language models perform marginally better than other tested language models and that combining visual and textual (ASR) evidence increases performance (MAP) over textual models alone.

## 5 Acknowledgements

Part of this work was based on work supported by Science Foundation Ireland under grant number 03/IN.3/I361. The support of the Informatics Directorate of Enterprise Ireland is also gratefully acknowledged. This material is based upon work supported by the IST programme of the EU in the project IST-2000-32795 SCHEMA. Part of this work was also supported by the European Commission under contract FP6-001765 aceMedia (URL: <http://www.acemedia.org>).

## 6 References

Blott S, Boydell O, Camous F, Ferguson P, Gaughan G, Gurrin G, Murphy N, O'Connor N, Smeaton AF, Smyth B and Wilkins P (2004). Experiments in Terabyte Searching, Genomic Retrieval and Novelty Detection for TREC-2004. *Draft Proceedings of the 13<sup>th</sup> Text REtrieval Conference (TREC 2004)*

Browne P, Gurrin C, Lee H, Mc Donald K, Sav S, Smeaton A F, and Ye J (2001). Dublin City University Video Track Experiments for TREC 2001. *Proceedings of the 10<sup>th</sup> Text REtrieval Conference (TREC 2001)*, Gaithersburg, Maryland, 13-16 November 2001.

Gaughan G, Smeaton AF, Gurrin C, Lee H and Mc Donald K (2003). Design, Implementation and Testing of an Interactive Video Retrieval System. *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR2003)*, Berkeley, CA, 7 November 2003.

Gauvain J L, Lamel L and Adda G (2002). The LIMSI Broadcast News Transcription System. *Speech Communication*, **37**(1-2):89-108, 2002.

Gurrin C, Lee H and Smeaton AF (2004). Físchlár @ TRECVID2003: System Description. *Proceedings of the 12<sup>th</sup> ACM International Conference on Multimedia 2004*, New York, NY, 15-16 October 2004, pp 938-939.

Lee H and Smeaton A F (2002). Designing the User-Interface for the Físchlár Digital Video Library. *Journal of Digital Information, Special Issue on Interactivity in Digital Libraries*, **2**(4), May 2002.

Manjunath B S, Ohm J R, Vasudevan V and Yamada A (2001). Color and Texture description. *IEEE trans. On Circuits and systems for video technology*, **11**(6), June 2001.

Schneiderman, H (2000). A Statistical Approach to 3D Object Detection Applied to Faces and Cars. Ph.D. Thesis, Carnegie Mellon University, CMU-RI-TR-00-06.

Smeaton A F, Gurrin C, Lee H, Mc Donald K, Murphy N, O'Connor N, O'Sullivan D, Smyth B and Wilson D (2004). The Físchlár-News-Stories System: Personalised Access to an Archive of TV News. *RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*, Avignon, France, 26-28 April 2004.

Westerveld T, Ianeva T, Boldareva L, de Vries A P and Hiemstra D (2003). Combining Information Sources for Video Retrieval: The Lowlands Team at TRECVID 2003. *TRECVID 2003*.

TRECVID2004 Guidelines (2004). Guidelines for the TRECVID 2004 Evaluation. Available online at URL: <http://www-nlpir.nist.gov/projects/tv2004/tv2004.html> (last visited October 2004)