TRECVid 2005 Experiments at Dublin City University

Colum Foley, Cathal Gurrin, Gareth Jones, Hyowon Lee, Sinéad McGivney, Noel E. O'Connor, Sorin Sav, Alan F. Smeaton, Peter Wilkins
Centre for Digital Video Processing & Adaptive Information Cluster
Dublin City University, Glasnevin, Dublin 9, Ireland
Alan.Smeaton@computing.dcu.ie

February 27, 2006

Abstract

In this paper we describe our experiments in the automatic and interactive search tasks and the BBC rushes pilot task of TRECVid 2005. Our approach this year is somewhat different than previous submissions in that we have implemented a multi-user search system using a DiamondTouch tabletop device from Mitsubishi Electric Research Labs (MERL). We developed two versions of our system one with emphasis on efficient completion of the search task (Físchlár-DT Efficiency) and the other with more emphasis on increasing awareness among searchers (Físchlár-DT Awareness). We supplemented these runs with a further two runs one for each of the two systems, in which we augmented the initial results with results from an automatic run. In addition to these interactive submissions we also submitted three fully automatic runs. We also took part in the BBC rushes pilot task where we indexed the video by semi-automatic segmentation of objects appearing in the video and our search/browsing system allows full keyframe and/or object-based searching. In the interactive search experiments we found that the awareness system outperformed the efficiency system. We also found that supplementing the interactive results with results of an automatic run improves both the Mean Average Precision and Recall values for both system variants. Our results suggest that providing awareness cues in a collaborative search setting improves retrieval performance. We also learned that multi-user searching is a viable alternative to the traditional single searcher paradigm, provided the system is designed to effectively support collaboration.

1 Introduction

This year Dublin City University participated in the TRECVid 2005 search and BBC rushes tasks only. We submitted four interactive runs and 3 fully automatic runs. Unlike our previous approaches to TRECVid experiments [6] where the system was evaluated using the traditional single user desktop interface, for this year's participation we evaluated a novel search environment, Físchlár-DT [15, 14], where two users search together for relevant shots using the DiamondTouch tabletop [7] a multi-user tabletop input device developed by Misubishi Electric Research Labs (MERL). We developed two versions of the system one with emphasis on increasing the awareness of users to the actions of the other user(Físchlár-DT Awareness) and the other with less of an emphasis on awareness and more of an emphasis on overall efficiency(Físchlár-DT Efficiency).

2 Interactive Search System

This year marks a departure from the standard DCU TRECVid search systems where the user interface was rendered by a web browser. In this years' system the user interface is rendered by the DiamondTouch multi-user table-top device. Our emphasis for the interactive search experiments this year is on exploring a multi-user single interface collaborative search for video retrieval. Our experiments this year had two participants working together to perform a collaborative search on the DiamondTouch device.

To explore issues of how users interact and how a collaborative search system can support this we developed two systems. The first is optimised for keeping each participant informed of the other's actions, which we refer to as the Awareness system. The second system we refer to as the Efficiency system and is designed to allow for rapid searching, with less regard given to keeping the other participant informed of the actions being undertaken and is closer to individual searching activity.

2.1 Architecture

The interactive video retrieval system we developed for this year's TRECVid no longer makes use of a browser or XML publishing framework as in previous years. This year the interface is displayed with the DiamondTouch. Because of this major changes were required to the generation of the interface which is now rendered through a custom built GUI. The server end has also been refined and simplified as the whole system can now be deployed onto a single machine as can be seen in Figure 1.

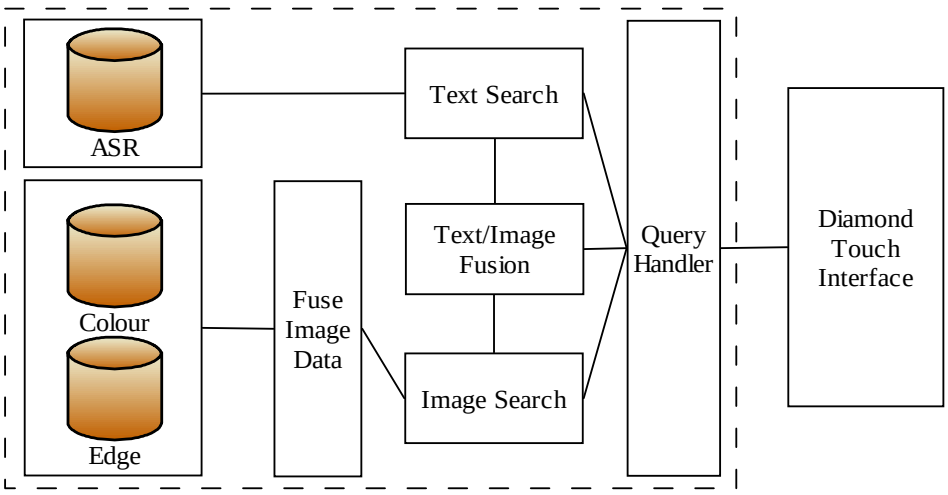

Figure 1: Físchlár-DT architecture

2.2 Retrieval and Weighting Scheme

Our TRECVid system for 2005 supported three types of query; text query, single image query and text and single image query. We also implemented a post query processing step which we refer to as augmentation. Each of these was handled by discrete components. The first of these to be described is text query searching.

Text queries were handled by our in house search engine Físrèal [8]. The text documents that were indexed were the ASR transcriptions for CNN and NBC. For the Chinese news we indexed the translations provided by CMU. Finally for the Arabic news sources we indexed the output of the Microsoft translation tool. Ranking of these shots was performed by an implementation of BM25 [12] adjusted for the smaller documents that comprise the TRECVid 2005 collection.

For image queries, we extracted the visual feature data from the collection making use of several feature descriptors based on the MPEG-7 XM. These descriptors were implemented as part of the aceToolbox, a toolbox of low-level audio and visual analysis tools developed as part of OUR participation in the EU aceMedia project [1]. Based on an evaluation we conducted on the performance of our visual descriptors based on last year's collection, we utilised both an Edge Histogram descriptor and a Local Colour Descriptor. These descriptors are defined as follows:

- An **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 edges are then categorised 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.

- A **Local Colour Descriptor (Colour Layout - CLD)** is a compact and resolution-invariant representation of colour in an image. The colour information of an image is partitioned in 64 (8x8) blocks; second, the representative colour of each block is determined by using the average colour in each block.

Our TRECVid system this year did not allow users to either search by individual features or adjust feature weights. We employed one set of static weights to combine the edge and colour data. These weights were determined by training on the 2004 collection. Each set of feature results was normalised based on score, then static weights were applied, followed by the combination into one list. This approach is further explained in [11].

Text and image combination queries are handled in a very similar way to image-only queries as has just been described. First we issue separate queries to both the text and image engines. The image engine combines its separate colour and edge data as previously described. We are left with two separate result lists, one for image and one for text, each of which are normalised. Again here we apply a set of static weights onto these normalised lists and combine the results. These weights were chosen based on experimentation with the 2004 TRECVid collection.

2.3 Augmented Post Query Processing

We implemented a basic version of post-query processing, which we've referred to as 'Augmentation'. Our augmentation process is based upon the experiments conducted by FXPAL in 2004 [3].

We first take the saved shots from the user and perform a basic bracketing operation, that is adding to the result list for each shot the preceding and following shot. Next we re-iterate through the shots that the user has selected (i.e. not the bracketed shots), and for each shot we take the keyframe and the associated ASR and run a text and image query. If no ASR is present we run an image-only query. We are left with an array of results, one for each shot that the user had saved. These arrays are normalised and combined into one result list. These results are then appended to the end of the bracketed shots to complete the list up to the 1000 shots allowed. In the event that no shots are submitted by a user for a topic, then no augmentation occurred.

2.4 DiamondTouch Hardware

The DiamondTouch is a multi-user tabletop input device developed by MERL. By using an overhead projector to render a PC display onto a tabletop surface users can use natural tendencies of reaching and touching in order to manipulate objects on-screen, thus creating a more natural interaction environment. The DiamondTouch hardware can handle up to four inputs simultaneously and allows inputs to be associated with a particular user through a receiver embedded in each user's chair.

2.5 DiamondSpin Software

DiamondSpin is a toolkit developed at MERL to support development of multi-user collaborative systems. It contains a well defined API of 30 Java classes and interfaces built using Java 2D with JMF (Java Media Framework) and JAI (Java Advanced Imaging). Unlike a traditional Cartesian coordinate system where there is a predefined position for displaying documents (keyframes in our case) DiamondSpin is based on a polar coordinate system where documents are displayed relative to one meaningful centre. The toolkit uses a real-time polar to Cartesian transformation engine which enables documents to be displayed at arbitrary angles and orientation. The framework consists of two key concepts:-

1. Translation of the origin of the Cartesian display (usually top left) to the centre of the tabletop
2. 3 Degrees of Freedom (DOF), `d`, α, β for each element on the display (see Figure 2)

2.6 User Interface

This year instead of a single user searching for a specific topic on a stand-alone PC, we used the DiamondTouch tabletop and DiamondSpin software to create a dual-user, collaborative system. The system we developed encourages interaction with the system to be both collaborative and much more physically natural by projecting the front-end of the system onto the DiamondTouch

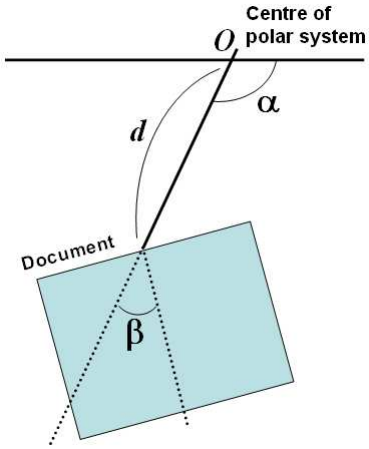

Figure 2: Document displayed at a distance d from the centre O at angle α. The document can then be rotated around its own centre using the β values

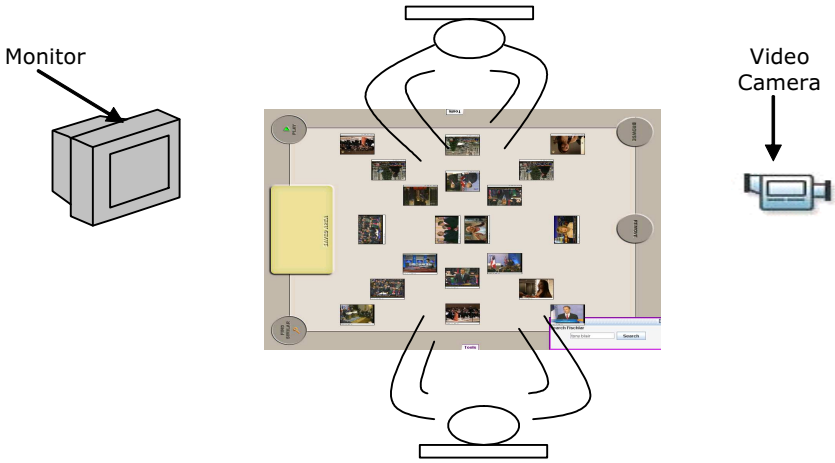

Figure 3: Typical Search Task Setup

Tabletop instead of using a traditional desktop PC. Each user will be able to manipulate on-table objects (in our case keyframes) directly using their fingers on the table. Our motivation behind this is to make the task of searching for shots, given a pre-specified time limit, a collaborative one and to see how effective collaborative searching can be.

2.6.1 Functionality

As previously stated, our system allows for a two-person collaborative shot/image search as shown in Figure 3. One video monitor is connected to the system to enable video playback of shots selected by either user. In addition, we capture user interaction with a CCTV video camera for later analysis.

Previous TRECVid systems have facilitated initial text-based querying and this feature is retained through the inclusion of a text search box in the interface. This search box comprises a text-box and "Search" button to activate the search. Only one of the users has control of this search box at any point in time, but it can be dragged around to the other user if wished. The text query is entered using a virtual keyboard supplied by the DiamondSpin software toolkit (the keyboard itself can be also dragged and rotated) to type in initial query terms. This is in keeping with the nature of the system - everything should be possible without the use of a peripheral device, just the use of fingers.

Once a query has been typed the "Enter" key is pressed, the keyboard disappears and the user

taps on the "Search" button. This sends the search terms to the underlying search server, which are then matched against the Automatic Speech Recognised (ASR) text of the video shot database and the top 7 matched shots are retrieved, along with the immediately preceding and succeeding shots for each of the retrieved shots. Subsequently the retrieved shots, represented by a keyframe are scattered around the tabletop surface with the more highly scored ones nearest the centre of the table.

The DiamondSpin software handles the display and orientation of shots on the tabletop and allows them to be resized, rotated and moved. It is possible to manipulate the retrieved keyframes in five ways: they can be Saved (i.e. deemed relevant by the user) - a yellow stamp stating "Saved" is then placed on the left hand corner and the keyframe surrounded by a yellow box; Removed from the tabletop surface, never to be displayed for this topic again; Played back on a separate, dedicated monitor to determine relevance; used for a Find Similar search (this function is described below) and Browsed showing the adjacent twenty (both previous and succeeding ten) shots. The ability to playback a shot will require some communication and agreement between users to avoid any conflicts that may arise (i.e. both users wanting to play a shot at the same time).

An important feature of the system is "Find Similar". Once selected, the system searches the database for other keyframes that are similar to a given keyframe (based on content similarity using the two MPEG-7 features), and the table will be populated with the top 20 most similar keyframes. Since selecting "Find Similar" means 20 more keyframes are added to the table, scattered all over, users are encouraged to clear up the table before taking this action.

2.6.2 Group Awareness

Whether using a distributed (remote) or co-located setting, the ability to be aware of what the other user is doing is an important consideration in collaborative interaction design. However, in comparison to a distributed setting, in a co-located tabletop setting group awareness comes more naturally since the collaborators are physically close around the table. For example, In User A's peripheral vision, she can inform herself of how actively User B is working. In a tabletop setting, more gestural physical/direct manipulations (e.g. dragging) tend to allow better group awareness than more symbolic manipulations (e.g. menu item selection, button clicking, short-cut keys), but with a reduced individual efficiency or power [9], and this is a trade-off the designer should decide on.

We developed two versions of our system to exploit this phenomenon, which we named "Awareness" and "Efficiency". Figure 4 and 5 show the Awareness and Efficiency systems respectively.

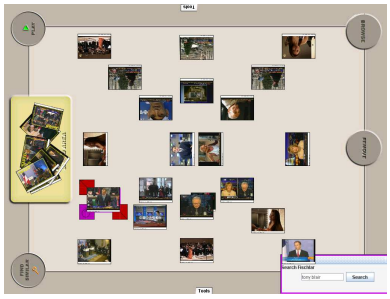

Figure 4: Físchlár-DT Awareness system

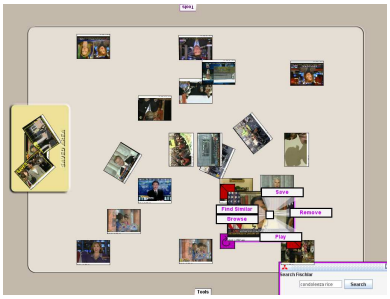

Figure 5: Físchlár-DT Efficiency system

The design of our Awareness system is such that it requires users to drag keyframes into certain areas of the table to invoke various functions. These areas were covered by a round image labeled with the function it invokes (see Figure 4). When a keyframe is placed over one of these images, an audible sound is emitted letting the user know that the function has been invoked. In using such a system, both users make their actions visibly and aurally known. Placing some of these functional images in the opposite user's "personal space" further increases awareness of the other user's actions. It may also require more coordination as the functional image on the opposite side of the table may be inconvenient to reach and so the keyframe might be passed to the other user to place on the required function area.

Each keyframe in the Efficiency version of the system can be double-tapped to display a context-sensitive pop-up menu on which the five possible actions (see Section 2.6.1) are displayed. This replaces the requirement to drag images onto certain areas of the table to invoke a function.

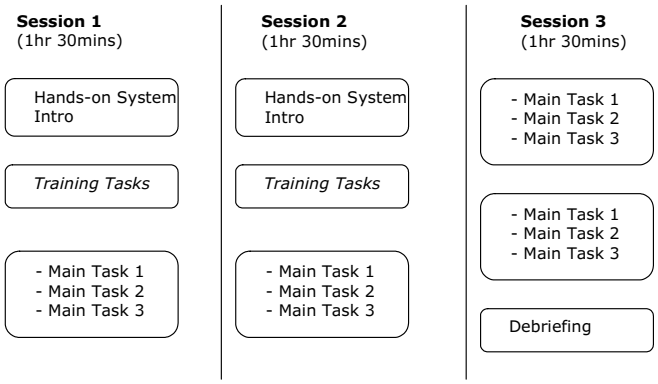

Figure 6: Organisation of Experiments

Figure 5 shows a table in the middle of searching, with some images saved and others still to be manipulated. Double-tapping on a keyframe is much more discreet than dragging it which results in users being much less aware of what the other is doing.

The most interesting and novel aspect of our system is the collaboration of multiple users and how they communicate in order to avoid conflict. Much of this information is captured by the CCTV video camera present in the room. By analysing CCTV footage we will determine the variation of the system that is most effective and most likely to provide sustainable results for a collaborative search task.

2.7 Experiment

Sixteen users (eight pairs) from within the Centre for Digital Processing participated in our experiments. Hence all of our users were experienced in the use of Information Retrieval systems, though the levels of this experience varied. All participants were expected to understand how to use single-user image and text search systems, such as those developed for previous TRECVid experiments and those available on the Internet e.g. Google.

The experiments were not as temporally flexible for participants as in previous years due to the nature of the system i.e. experiments had to be conducted sequentially as opposed to in parallel and at the participant's convenience. It was necessary to timetable the experiments in order to ensure participants were available for the timeslots allocated to them. We created sessions of one hour and a half and each pair completed three such sessions. These were organised as follows:

Each user was sent a pre-experiment questionnaire via email to be completed prior to starting Session 1. Along with this, a link to an online personality questionnaire was sent to be completed at the user's earliest convenience. They then brought these with them when they began their first session. Session 1 involved the use of one version of the collaborative system. The evaluator gave each pair an introduction to the system. The users then completed some training tasks comprised of sample topics and when users felt comfortable using the system, they completed three of the main tasks. Each pair had to complete each task within a period of 10 minutes. Session 2 involved the use of the alternative system and the layout of the session followed that of Session 1. Session 3 involved completing three main tasks on both versions of the system. A system evaluator (who was also one of the system developers) was present for the duration of each of the sessions to answer any questions or handle any unexpected system issues.

Pairs of users did not complete the sessions consecutively. This was due to the fact that the sessions were already quite long and so their concentration would fade by the end of a session. In addition to this, users were offered the chance to take a break in between tasks should they feel fatigued.

Unfortunately due to time restrictions, we were unable to undertake Session 3 before the interactive results submission deadline. This session was hence conducted after the submission deadline and the results from the three sessions combined. These results are included in section 2.7.3

2.7.1 Experimental Design Methodology

Similar to last year, this year's interactive search task guidelines included guidelines for the design of user experiments (TRECVid guidelines, 2005). 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 (Awareness, Efficiency) using 24 topics and either 8, 16 or 24 searchers, each of whom searches 12 topics. A searcher does not see the same topic more than once and a searcher completes all work on one system variant before beginning any work on the other variant.

We followed the guidelines for evaluating instead of single users, 8 pairs of users (sixteen participants in total) searching 12 topics using 2 system variants (we treated each pair as one user for the purposes of the design methodology). The selection of these pairs was based on who we thought would work well together (i.e. friends, people that worked together on certain projects, members of the same lab etc). Pairs of searchers were assigned randomly and the order of topic presentation for a given system and searcher-pair 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 searcher-pair and topic and the gathering of some information about interactions.

2.7.2 Experimental Procedure

A pair of users were brought into a room that contained the Físchlár-DT system. They each gave their completed pre-experiment questionnaire and personality test result and then sat at opposite sides of the DiamondTouch tabletop. After initiating the system on the tabletop we explained how the system worked and how to use all of its features. We then conducted a series of test searches until the users felt comfortable using the system. Following these, the main search tasks began.

Users were told the topics to search for, and were then handed a physical printout of these topics. Users also had the option of displaying seed images (i.e. examples of images relevant to the topic currently being searched for) in order to initiate an image search. As previously stated, users were given 10 minutes for each topic and all relevant images were placed in the saved area (the procedure for doing this differed in each version of the system - see section 2.6.2 for details of this). After each pair completed the three sessions, they were asked to each complete a post-experiment questionnaire.

Each individual's interactions were logged by the system, as well as each session being captured by a video camera placed next to the users in the room. The purpose of this was to capture the coordination, cooperation and effectiveness of each of the pairs for future analysis. The results of users' searching (i.e. shots in saved area) were collected and from these results eight runs were submitted to NIST for evaluation.

2.7.3 Submitted Runs

For our interactive search experiment we submitted 4 official runs, two runs for each of the system variants, one consisting of the original search results (dcu.videoSearchResult_Efficiency & dcu.videoSearchResult_Awareness) and another consisting of the original search results augmented with the results from a modified automatic run as described in Section 2.3 (videoSearchResult_Efficiency_Augmented, videoSearchResult_Awareness_Augmented). Due to time restrictions our submitted runs consisted of 8 groups completing 6 topics, 3 on each system. We have since completed the other half of the experiment and have also corrected a bug which we had noticed with our original system. We present below the results from our combined runs: 8 groups with each group performing 12 topics, 6 topics on each system. Therefore on each system, each topic was run by two groups.

Table 1: Interactive Experiment Results

Run Name	MAP	P@10	Recall
Físchlár-DT Awareness	0.1529	0.7167	0.0685
Físchlár-DT Efficiency	0.1372	0.6042	0.0673
Físchlár-DT Awareness_Augmented	0.2100	0.7625	0.1671
Físchlár-DT Efficiency_Augmented	0.1862	0.6917	0.1605

Overall, the Awareness system outperforms the Efficiency system, with Mean Average Precision (MAP) and Recall values of 0.1529 & 0.0685 respectively for the original (non-augmented runs) compared with figures of 0.1372 & 0.0673 for the Efficiency based system. This result is interesting; due to the symbolic interaction style of the Efficiency system we had originally expected it to outperform the Awareness system in the intensive manner of TRECVid search tasks. The Awareness based system on the other hand, with its "dragging" interaction metaphor, was designed to be more user friendly with less emphasis on efficient completion of the search task. The augmented runs, where the original user results were expanded through automatic procedures performed significantly better than their respective original user runs with substantial increases in MAP and Recall for both systems. During our experiments we noticed a number of software issues which may have impeded performance of both systems to a certain extent. Due to the nature of these issues it is difficult to derive a definitive result as to the best system design until these have been resolved.

2.8 Automatic Runs

This year DCU again participated in the automatic and manual search tasks . We submitted three runs, two automatic (*dcu.automatic.text.img.auto.weight* and *dcu.automatic.text.only*) and one manual (*dcu.manual.text.img*). One of the automatic runs was text only, whilst the others incorporated both text and image data. We identified a minor bug in our text only run which has lead to a minor MAP adjustment. Our results are presented in Table 2.

Table 2: Automatic and Manual Results

Run Name	MAP	Recall
dcu.automatic.text.only	0.046	1247
dcu.automatic.text.img.auto.weight	0.078	1209
dcu.manual.text.img	0.081	1648

All of the text results used the text index explained in Section 2.2. The text results were stopped with the SMART stopword list which was extended to include the follow TRECVid specific keywords: "find, additional, shots, scenes, pictures, containing, including, showing, lots, groups, multiple, partly, partially, visible". We also applied a weighted windowing scheme, where five shots were taken either side of the returned shot, and their scores were multiplied by the following weights: "1.0, 0.9, 0.75, 0.5, 0.25". Once all results were windowed the results were re-ranked and formed the text results.

Our image results were weighted using a derivative of our image retrieval work detailed in [16]. However this weighting scheme still has several issues to address and requires further refinement. For our two colour features that we used (Edge Histogram and Colour Layout as defined in Section 2.2) it produced approximate weights of 0.6 and 0.4 respectively.

Combination of text and image data was performed in a similar fashion as explained in Section 2.2, with text and image data sharing equal weighting.

Preliminary analysis of our results highlights several interesting issues, primarily the effectiveness of image and text data sources when compared to the 2004 collection. For the 2004 collection, if we apply the same text weighting scheme as used for 2005, we achieve a MAP of 0.072, whereas in 2005 the very same approach yielded a MAP of 0.046. Conversely for an image-only run using colour and edge data combined with a static weight scheme, we achieve in 2004 a MAP of 0.028, whereas using the exact same scheme in 2005 yields a MAP of 0.069. An obvious early conclusion is that given the nature of the collection and the topics provided that the two collections are not as comparable as would initially seem as they are both comprised of broadcast news. However a closer examination of our results indicates that our scores are being dominated by a handful of topics that perform very well, whilst the rest are negligible. Therefore it may be the case that this year consisted of a couple of topics that were very conducive to our approach, however further investigation will need to be conducted.

3 BBC Rushes Task

3.1 Summary of Task Definition and Our Approach

DCU also took part in the BBC rushes pre-task as part of TRECVid 2005 and our effort was to explore how users might use video *objects* as part of retrieval, as opposed to using whole keyframes. Our thesis here was that there are certain types of information need which lend themselves to expression as queries where objects – which could be cars, people, trees, the Eiffel tower, horses, or a pair of shoes – form a central part of that and we wanted to investigate how true this was for very unstructured video such as rushes.

The data used and the overall task is described in detail elsewhere in these proceedings but in summary we have 50 hours of rushes video provided by BBC [1] and the task for participants in this initial track is to explore how we could develop techniques to automatically analyse such data and build systems which allow users who know nothing about the content of the data to navigate through it with some information need in mind. Rushes is a term used to refer to raw video footage which is unedited and contains lots of redundancy, overlap and "wasted" material. Shots tend to be much longer than in post-produced video and it generally contains a lot of re-takes, bloopers and content where nothing much happens.

In previous work reported elsewhere [5] we developed a video retrieval and browsing system which allowed users to search using the text of closed captions, using the whole keyframe for locating keyframes similar in terms of colour, texture and edges, and using the occurrence (or non-occurrence) of a set of pre-defined video objects. The content used was several seasons of the Simpsons TV series and the video objects corresponded to the faces of the 10 major characters in the series, Homer, Bart, Marge, etc. We evaluated the ways in which different video retrieval modalities (text search, image search, object search) were used and we concluded that certain queries can benefit from using object presence as part of their search, but this is not true for all query types. In retrospect this may seem obvious but we are all learning that different query types need different combinations of video search modalities, something best illustrated in the work of the Informedia group at ACM Multimedia in 2004 [10].

In trying to move from object detection and retrieval on synthetic video (cartoons, the Simpsons) to working on natural video we are of course faced with the problem of having to do object segmentation. This has been a focus of much research and attention as it is the basis for MPEG-4 video coding, whatever other benefits it may offer to video retrieval or interaction with video. Object segmentation in video is hard because objects can deform and turn, cameras can move, objects can become occluded when other objects move in front of them, lighting conditions can change, and so on.

Despite the inherent difficulties of object segmentation we have developed and implemented a system which can support object-based matching of video shots. We addressed the requirement for object segmentation by developing a semi-automatic segmentation process [13]. We have evaluated this as a search mechanism informally in-house but we have not yet evaluated it directly in comparison with other retrieval modalities. The BBC rushes task offers us the opportunity to use our object-based video retrieval technique as part of an integrated system for searching and browsing BBC rushes and that is what we have done here.

Using the existing system as a starting point we built a system for the BBC rushes initial task which supports whole keyframe based retrieval and also supports object based retrieval. Keyframe retrieval can use a number of query keyframes as the query and object based retrieval can also use a number of exemplar objects as the query. These two video retrieval approaches can also be combined into the one query for retrieval. In the next section we give an outline of the system developed.

3.2 System Description

Our system begins by analysing raw video data in order to determine shots. For this we use a standard approach to shot boundary determination, basically comparing adjacent frames over a certain window using low-level colour features in order to determine boundaries [4]. From 50 hours of video we detected 8,717 shots, or 174 keyframes per hour, much less than for post-produced video such as broadcast news. For each of these we extracted a single keyframe by examining the whole shot for levels of visual activity using features extracted directly from the video bitstream.

[1] BBC 2005 Rushes video is copyrighted. The BBC 2005 Rushes video used in this work is provided for research purposes by the BBC through the TREC Information Retrieval Research Collection.

The rationale for this is that the "usual" approaches of choosing the first, last or middle frames as the keyframe would be quite inappropriate given the amount of "dead" time there is in shots within rushes video. Much of the unusable video footage in rushes is there because the camera is left running while the main action of the shot is prepared and then takes place. In rushes footage the camera is left running in order to ensure the action, whatever that may be, is not missed. Thus an approach to keyframe selection based on choosing the frame where the greatest amount of action is happening seems to make sense, although we admit that this is a bit of a black art and is certainly a topic for further investigation.

Each of the 8,717 keyframes were then examined to determine if there was a single main object present. For such keyframes that object was semi-automatically segmented from its background using a segmentation tool we had developed and used previously [2]. This is based on performing an RSST segmentation using homogeneous colour, yielding many small regions for each frame. Our semi-automatic segmentation tool then requires a user to scribble on-screen using a mouse to indicate a region inside, and a region outside the dominant object. This process is very quick for a user to perform, requires no specialist skills and yielded 1,210 such objects.

Once the segmentation process was completed we proceeded to extract features for keyframes in two ways:

1. Global MPEG-7 colour and texture features were extracted for all keyframes;
2. Dominant colour, texture browsing and shape compactness MPEG-7 features were extracted for all segmented objects;

This effectively resulted in two separate representations of each keyframe/shot. We then pre-computed two $8,717 \times 8,717$ matrices of keyframe similarities using colour and texture for the whole keyframe and three $1,210 \times 1,210$ matrices of similarities between those keyframes with segmented objects using colour, texture and shape.

For retrieval or browsing this or any other video archive with little metadata to describe it, we cannot assume that the user knows anything about its content since we assume it is not catalogued in the conventional sense. So, in order to kickstart a user's retrieval we ask the user to locate one or more images from outside the system using some other image searching resource. The aim here is to find one or more images, or even better one or more video objects, which can be used for searching. In our experiments our users use Google image search to locate such external images but any image searching facility could be used. Once external images are found and downloaded they are analysed in the same way as keyframe in terms of colour and texture for the whole image and the user is also allowed to semi-automatically segment one object in the external image if s/he wishes.

When these seed images are ingested into our system the user is asked to indicate which visual characteristics make each seed image a good query image — colour or texture in the case of the whole image or colour, shape or texture in the case of segmented objects in the image. Once this is done the set of query images is used to perform retrieval and the user is presented with a list of keyframes from the archive. For keyframes where there is a segmented object present (1,210 of our 8.717 keyframes) the object is highlighted when the keyframe is presented. The user is asked to browse these keyframes and can either play back the video, save the shot, or add the keyframe (and its object, if present, to the query panel) and the process of query-browse can continue until the user is satisfied. A sample screen taken from the middle of a search is shown as Figure 7 where there are 4 query images, the first, second and fourth with segmented objects, 6 pages of results and 4 saved keyframes, and the overall architecture of our system is shown as Figure 8.

3.3 Experiments

We used two versions of the system described above to explore how useful video objects are in video browsing and search. System A supports image-image similarity based on whole keyframes, whilst system B supports object-object similarity as well as whole keyframe matching. At the time of writing a small user experiment is being run to compare the performance of the two systems for a small number of search tasks. Preliminary results from this and some analysis will be presented at the TRECVid workshop.

Acknowledgements

The authors gratefully acknowledge the support of Mitsubishi Electric Research Labs (MERL). This work is part-funded by the Irish Research Council for Science Engineering and Technology

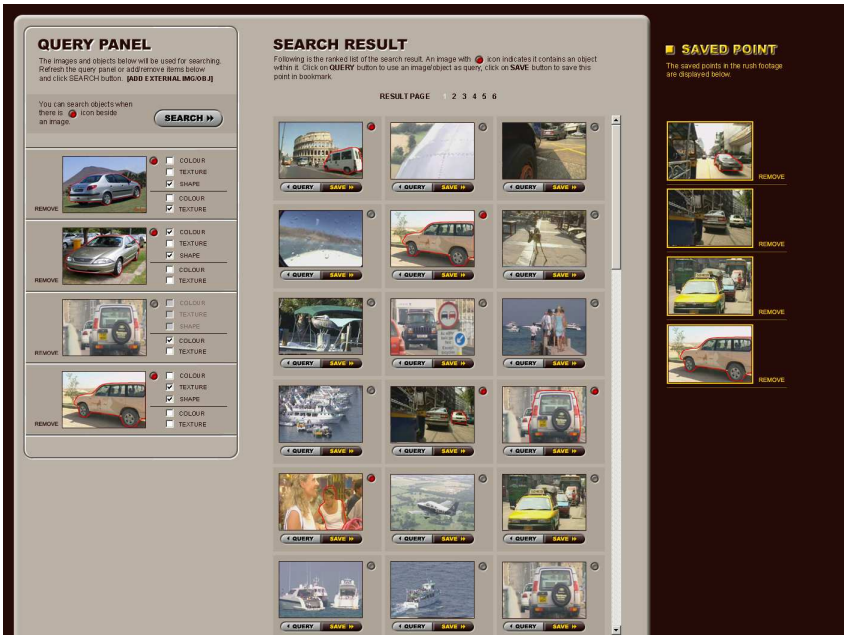

Figure 7: Sample screen from our BBC Rushes search system.

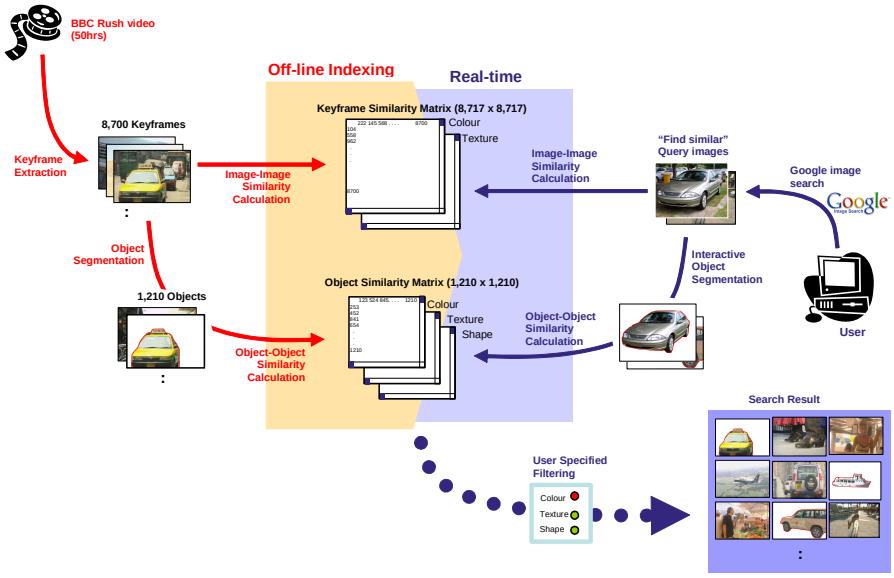

Figure 8: System Architecture for our BBC Rushes search system.

and is partially supported by Science Foundation Ireland (grant 03/IN.3/I361).

References

[1] The AceMedia project, available at http://www.acemedia.org.

[2] T. Adamek and N. O'Connor. A Multiscale Representation Method for Nonrigid Shapes With a Single Closed Contour. In *IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on Audio and Video Analysis for Multimedia Interactive Services*, 2004.

[3] J. Adcock, A. Girgensohn, M. Cooper, T. Liu, L. Wilcox, and E. Rieffel. FXPAL experiments for TRECVID 2004. In *Proceedings of TRECVID 2004*, November 2004.

[4] P. Browne, C. Gurrin, H. Lee, K. M. Donald, S. Sav, A. Smeaton, and J. Ye. Dublin City University Video Track Experiments for TREC 2001. In *TREC 2001 - Text REtrieval Conference*, 2001.

[5] P. Browne and A. F. Smeaton. Video Retrieval Using Dialogue, Keyframe Similarity and Video Objects. In *ICIP 2005 - International Conference on Image Processing*, 2005.

[6] E. Cooke, P. Ferguson, G. Gaughan, C. Gurrin, G. J. F. Jones, H. L. Borgne, H. Lee, S. Marlow, K. McDonald, M. McHugh, N. Murphy, N. E. O. Connor, N. O'Hare, S. Rothwell, A. F. Smeaton, and P. Wilkins. TRECVID 2004 Experiments in Dublin City University. In *Proceedings of TRECVID 2004*, November 2004.

[7] P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In *ACM UIST'01*, pages 219–226, 2001.

[8] P. Ferguson, C. Gurrin, P. Wilkins, and A. F. Smeaton. Físréal: A Low Cost Terabyte Search Engine. In *Proceedings of ECIR 2005*, 2005.

[9] C. Gutwin and S. Greenberg. Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness. In *ACM CSCW'98*, pages 207–216, 1998.

[10] A. Hauptmann and M. Christel. Successful Approaches in the TREC Video Retrieval Evaluations. In *Proceedings of ACM Multimedia 2004*, 2004.

[11] K. McDonald and A. F. Smeaton. A Comparison of Score, Rank and Probability-based Fusion Methods for Video Shot Retrieval. In *Proceedings of CIVR 2005*, 2005.

[12] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[13] S. Sav, H. Lee, A. Smeaton, N. O'Connor, and N. Murphy. Using Video Objects and Relevance Feedback in Video Retrieval. In *Proceedings of SPIE (SPIE, Bellingham, Wa)*, 2005.

[14] A. F. Smeaton, C. Foley, C. Gurrin, H. Lee, and S. McGivney. Collaborative Searching for Video Using the Físchlár System and a Diamondtouch Table. In *The 1st IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, 2006.

[15] A. F. Smeaton, H. Lee, C. Foley, S. McGivney, and C. Gurrin. Físchlár-Diamondtouch: Collaborative Video Searching on a Table. In *Proceedings of SPIE Electronic Imaging - Multimedia Content Analysis, Management, and Retrieval*, 2006.

[16] P. Wilkins, P. Ferguson, A. F. Smeaton, and C. Gurrin. Text Based Approaches to Content-Based Image Retrieval. In *Proceedings of EWIMT 2005*, 2005.