Video Retrieval using Search and Browsing

Daniel Heesch, Peter Howarth, João Magalhães, Alexander May,
Marcus Pickering, Alexei Yavlinsky, Stefan Rüger

Department of Computing, South Kensington Campus, Imperial College London, SW7 2AZ, UK
{dh500, pdh101, jmag, adm00, mjp03, agy02, s.rueger}@imperial.ac.uk

Abstract. We describe our experiments for the shot boundary detection, high-level feature extraction, search and story boundary detection tasks. In the shot boundary detection task, we employ our proven method based on the calculation of distances between colour histograms of frames over a range of timescales. In the feature detection task we confined our effort to the basketball feature. For the search task, we test a complete redesign of last year's interface. Like last year, content-based search is complemented with NN^k browsing functionality. We compare performance of two interactive runs, one including search and browsing, the other including search only. Overall search performance falls behind that of last year, a result which we would like to attribute to the stronger emphasis on motion in this year's search task.

1 Introduction

The retrieval system we use for the search tasks is functionally similar to that used for TRECVID 2003 [4] but with a tighter integration of search and several types of browsing. Both search and browsing are based on global and semi-global image features. Unlike last year, we have included a suite of texture features but have made no attempt to model the motion aspect in video. Two interactive runs are carried out to test whether the use of the NN^k network significantly improves performance above and beyond the performance obtained for search only. The paper is structured as follows. Section 2-5 describe, respectively, the experiments and results for the shot boundary detection task, the search task, the feature detection task and the story boundary detection task.

2 Shot boundary detection task

2.1 System

The video shot boundary detection algorithm is broadly based on the colour histogram method. The colour histograms for consecutive frames are compared and a shot change is declared if their difference is greater than a given threshold. This method is extended based on the algorithm of Pye et al [11] for detection of gradual transitions that take place over a number of frames, and for rejection of transients such as the effect of a flash-bulb. In order to determine the start and end points for gradual transitions, we employ a method similar to that described by Zhang [15], in which a lower threshold is used to test for the start and end of a gradual transition. Our system is largely unchanged from the last two years, and more details can be found in our TREC 2002 proceedings paper [10].

2.2 Experiments

We performed ten shot boundary detection runs. The first four runs, Imperial-01 – Imperial-04 were carried out keeping the low threshold, T_4, constant, and reducing the high threshold, T_{16}. In runs Imperial-05 – Imperial-08, the low threshold was increased and the same 3 values for the high threshold were used again. The last two runs, Imperial-09 and Imperial-10 used a value for the low threshold falling between the first set of runs and the second set, and a high and low value respectively for the high threshold. The thresholds for these last two runs were determined empirically from previous year's TREC results.

	All		Cuts		Gradual			
	Recall	Prec	Recall	Prec	Recall	Prec	F-Recall	F-Prec
Imperial-01	0.717	0.883	0.835	0.889	0.469	0.861	0.856	0.153
Imperial-02	0.729	0.864	0.831	0.871	0.513	0.840	0.871	0.160
Imperial-03	0.736	0.842	0.828	0.851	0.544	0.813	0.877	0.166
Imperial-04	0.738	0.799	0.823	0.810	0.559	0.766	0.873	0.172
Imperial-05	0.814	0.846	0.895	0.885	0.644	0.748	0.773	0.560
Imperial-06	0.836	0.817	0.895	0.865	0.713	0.711	0.764	0.566
Imperial-07	0.851	0.779	0.895	0.836	0.758	0.667	0.748	0.567
Imperial-08	0.854	0.731	0.892	0.783	0.775	0.629	0.755	0.551
Imperial-09	0.816	0.834	0.883	0.869	0.673	0.751	0.821	0.412
Imperial-10	0.830	0.801	0.881	0.845	0.723	0.707	0.816	0.419
TRECVID Median	0.756	0.842	0.888	0.869	0.485	0.683	0.701	0.662

Table 1. Shot boundary detection task – results summary

2.3 Results

We show the results for our ten shot-boundary detection runs in Table 1. There was a predictable trade-off between recall and precision as the thresholds were altered. Runs 5, 6 and 9 gave particularly good points at the top right of the precision-recall graphs.

In addition we measured the clock time for each run to get an indication of the efficiency. A run consisted of processing all of the 12 test files by a single system variant. The only differences between each of the runs were the threshold settings. Therefore the total time for each run was the same, 9660 seconds. The timing runs were performed on a Pentium 4, 2.8GHz.

3 Search task

3.1 Graphical user interface

We have consolidated the paradigms of text-based search, content-based search with relevance feedback, NN^k browsing, temporal browsing and historical browsing into a unified interface. By providing tight integration of techniques and a rich set of user interactions, we aimed to equip the user with substantial navigational power.

The user-interface of the browser was developed using a user-oriented approach. At each stage we considered how the features of the interface would enhance the browsing process for the user. Several key concepts of HCI were maintained throughout the design process:

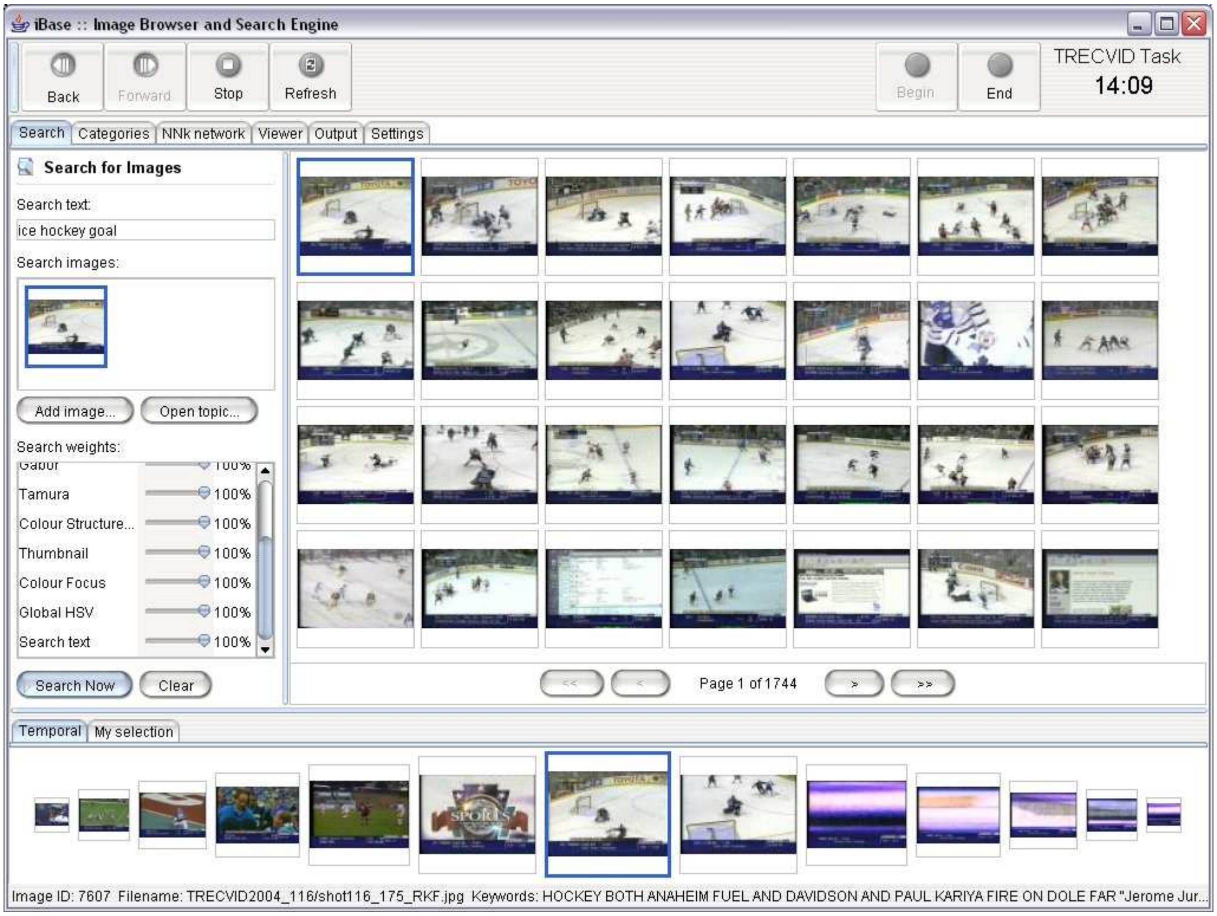
Consistency: both within the browser interface and with recent standards in human-computer interfaces. For example, HCI standards would suggest that left-clicking on an image selects it and right-clicking would provide a context-aware popup menu with the available actions that can be performed.

Responsiveness: the user should at no point be presented with an unresponsive interface. We must ensure that there is always be some processing power available to servicing the interface. Similarly, if the browser is occupied performing time-consuming computation or retrieval then the user should be able to continue with other actions, and have the option to cancel the task.

System progress feedback: when a time-consuming task is taking place, such as computation or retrieval of data from the network we must ensure that the user is kept informed of progress.

The interface is designed upon the notion of an “image of interest”. A tabbed layout is used to separate the different methods, allowing the user to browse in different dimensions with respect to a particular image. The current “image of interest” is displayed with a blue border in each panel. The user selects an image as “image of interest” simply by clicking on it. This updates all other panels to show this image and hence maintain consistency across all the views.

Figure 1 shows the interface layout with the search panel selected. The user formulates a query using the left hand panel. Below we outline the stages of the searching process:



TRECVID Task 14:09

Search text: ice hockey goal

Search images:

Search weights:

Gabor 100%
Tamura 100%
Colour Structure... 100%
Thumbnail 100%
Colour Focus 100%
Global HSV 100%
Search text 100%

Page 1 of 1744

Image ID: 7607 Filename: TRECVID2004_116/shot116_175_RKF.jpg Keywords: HOCKEY BOTH ANAHEIM FUEL AND DAVIDSON AND PAUL KARIYA FIRE ON DOLE FAR "Jerome Jur...

Fig. 1. Display of Initial search result with panel for temporal browsing

Initial query formulation: query text and images are loaded as per topic and modified as necessary. Features are initially equally weighted but can be modified manually by using the appropriate sliders.

Browsing of search results: results are returned in a line-by-line page-wise fashion that is common for search engines, with the most relevant image in the top-left corner, and the least relevant in the bottom-right corner. This traditional approach is simple, intuitive and makes maximum use of the available space, helping the user to assess the relevance of many images quickly and use them in the search query if required. In this way, we maximise feedback from the user in the searching process.

Relevance feedback: the user is able to modify the query text, images or feature weightings and search again to improve results. To assist the user in finding the optimum feature weight set, we have provided an automatic relevance feedback technique where the user simply marks which images in the search results are relevant. The system then finds the feature weights for which these images are ranked highest in the results. In this way, the system encodes the users notion of a "relevant" image within the feature descriptor weightings, and there is an improvement in the quality of search results.

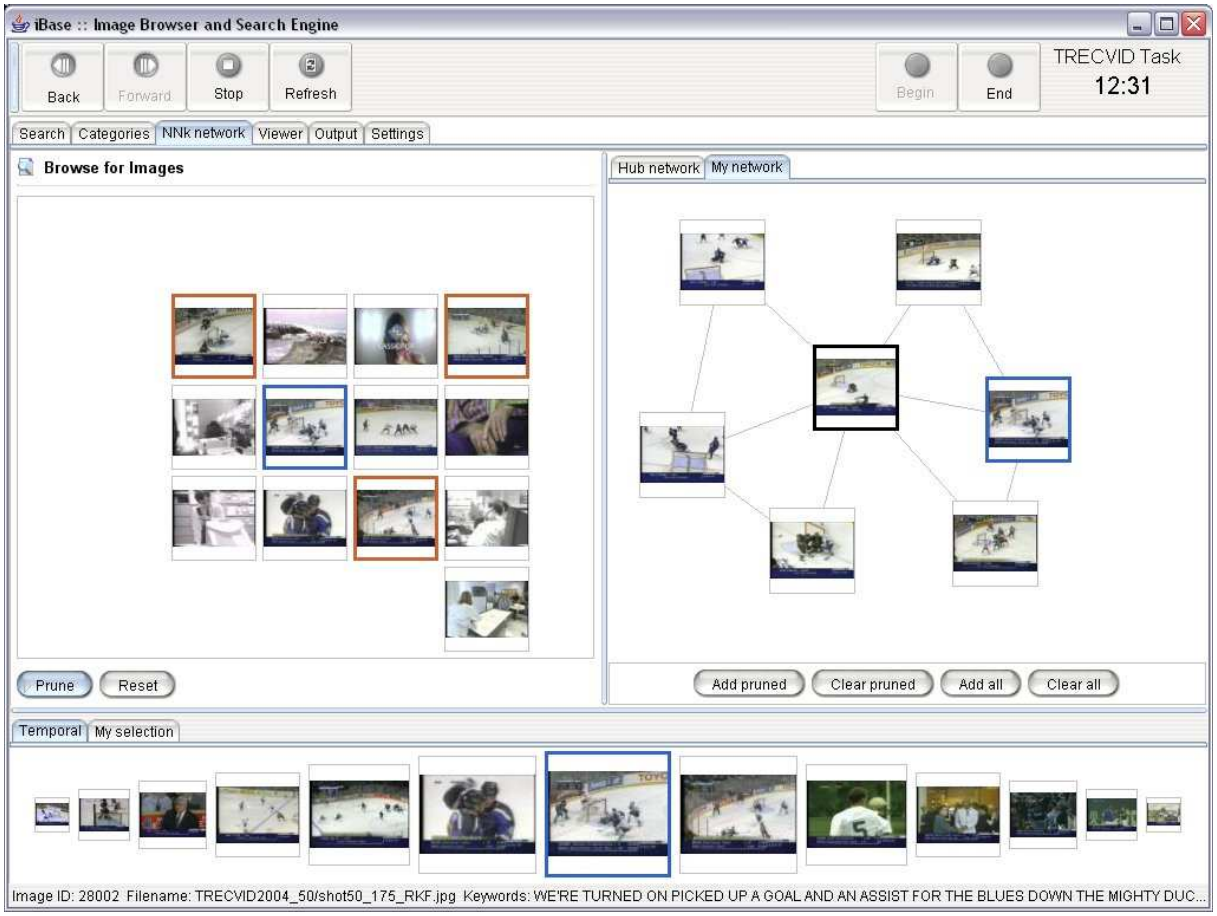


Image ID: 28002 Filename: TRECVID2004_50/shot50_175_RKF.jpg Keywords: WE'RE TURNED ON PICKED UP A GOAL AND AN ASSIST FOR THE BLUES DOWN THE MIGHTY DUC...

Fig. 2. Constructing your own NN^k network to gather more relevant shots

Figure 2 shows the interface with the NN^k browsing panel selected. This enables the user to navigate across the NN^k network. We provide the 36 most highly-connected "hub" images in the network as a useful starting point. When the user selects an image, either from the hub images or anywhere else in the browser, the nearest neighbours are shown on the right-hand 'pruning' panel. The user can assess the neighbours for relevance and add relevant images to a custom network which shows the neighbour relationship between images. All previously assessed images do not appear again in the browsing process. The user can select a new "image of interest" from the custom network, and the pruning panel is updated to display that images neighbours. By iteratively pruning neighbours in the custom network, the user can explore an image collection to find relevant images quickly and efficiently. The high level of user-feedback makes NN^k browsing well suited to complex high-level queries which are hard to describe in keywords or using feature descriptors from query images. Hence it is a perfect complement to the traditional searching techniques.

Temporal browsing is provided using a fisheye visualisation at the bottom of the window. This allows the user to effectively combine temporal browsing with searching or NN^k browsing. Once a user has located a relevant image, browsing along the temporal neighbours of that image often yields further relevant images.

As well as allowing the user to interleave searching and browsing methods via the image-of-interest paradigm, we have integrated the different functionality in other ways. We have used context-aware popup menus to provide this level of integration. By right-clicking on an image in any of the panels, the user is presented with operations that can be performed to an image under that context. For example, the user can right-click on an image in the NN^k browsing panel to add it to a content-based search query. Similarly, the user can right-click on an image in the NN^k browsing pruning panel, in order to use the feature weight set which corresponds to that image in the nearest neighbour network in a content-based search. The user can also add images from any panel to a custom collection of images, which is shown on the bottom of the screen by selecting the "My Selection" tab. This further enhances the ability to interleave different searching and browsing methods and gives the user added flexibility in the browsing strategy.

Each panel in the browser uses its own thread to perform time-consuming tasks which ensures that the event-dispatching thread is available for servicing the interface. Thus the user is in complete control and can "Stop" and "Refresh" the current view at any time. The user is kept informed of system progress using a status bar showing the total number of images remaining to be processed. These features are consistent with other browsers, such as the familiar web-browsers.

3.2 Features

For the search task we used nine low-level texture and colour features as well as the text from the LIMSI transcripts. Like last year, we cut of the bottom of each image as this often contained a news line. To capture local image information, we partition the images into tiles and obtain features from each tile. The final feature vector consists of a concatenation of the feature vectors for individual tiles. For features that we used in last year's TRECVID, we refer the reader to our proceedings paper.

Tamura Features We compute the three Tamura features coarseness, contrast and directionality as proposed in [12] for each of $9x9 = 81$ tiles. For each pixel we compute the values of each of the three features and compute for each tile a 3D histogram. See our paper [6] for additional details and evaluation results.

Co-Occurrence Feature In 1979 Haralick [3] suggested the use of grey level co-occurrence matrices (GLCM) to extract second order statistics from an image. They have subsequently been used successfully for texture classification. The GLCM of an image is defined as a matrix of frequencies at which two pixels, separated by a certain vector, occur in the image. The distribution in the matrix will depend on the angular and distance relationship between pixels. Varying the vector used allows the capturing of different texture characteristics. Once the GLCM has been created, various features can be computed from it. These have been classified into four groups: visual texture characteristics, statistics, information theory and information measures of correlation [3, 2].

Using the results of our recent evaluation [6] we chose the following configuration for our feature:

- The original image was split into 9×9 non-overlapping tiles and the feature run for each of these;
- The colour image was quantised into 32 grey levels;
- 16 GLCMs were created for each image tile using vectors of length 1, 2, 3, and 4 pixels and orientations 0, $\pi/4$, $\pi/2$ and $3\pi/4$;
- For each normalised co-occurrence matrix $P(i, j)$ we calculated a homogeneity feature H_p,

$$H_p = \sum_i \sum_j \frac{P(i,j)}{1 + |i - j|}$$

This feature was chosen as it had performed consistently well in previous evaluations.

Gabor Filter One of the most popular signal processing based approaches for texture feature extraction is the use of Gabor filters. These enable filtering in the frequency and spatial domain. A range of filters at different scales and orientations allows multichannel filtering of an image to extract frequency and orientation information. This can then be used to decompose the image into texture features.

Our implementation of the Gabor filter is based on [7]. To each image we apply a bank of 4 orientation and 2 scale sensitive filters that map each image point $I(a, b)$ to a point in the frequency domain:

$$W_{nm}(a, b) = \int I(a, b) g_{mn}(x - a, y - b) dx dy$$

The feature consists of the mean and standard deviation of the modulus of W_{nm}. Since different filters produce outputs in different ranges, we normalize both mean and standard deviation by dividing by the respective standard deviation obtained for the entire database. We use a tiling of $7x4$ based on experiments previously carried out on the Corel collection [6].

Global and Local HSV Feature These two features are the same as in the previous year (see [4]).

Thumbnail Feature This feature is obtained by scaling down the original image to 44×27 pixels. For details see [4].

Convolution Filter This feature is based on Viola's method [13], for details see [4].

Colour Structure Descriptor Details can be found in [8] and [4].

Text Feature The text feature was derived by aligning LIMSI ASR transcripts, closed caption data, and OCR data provided by Alexander Hauptmann's group at CMU. We smoothed each keyframe's text by including terms from the previous and the subsequent frames and duplicating the text of the frame itself. A full text index was built using Lucene and queries formed from the XML data supplied with each query.

Bag-of-Words Feature Using the textual annotation obtained using the above procedure, we computed a bag-of-words feature consisting for each image of the set of accompanying stemmed words (Porter's algorithm) and their weight. This weight was determined using the standard tf-idf formula and normalised so that the sum of all weights is 1. As this is a sparse vector of considerable size (the number of different words) we store this feature in the form of (weight, word-id) pairs, sorted by word-id.

3.3 Distances and Normalisation

In order to compare two images in the data-base we use distances of their corresponding descriptors. For these we use the L_1-norm throughout. Some of the distances already exhibit a natural normalisation, for example when the underlying features are normalised (eg, HSV colour histograms), others do not (eg, the colour structure descriptor). As the distances under different descriptors are going to be combined, we normalise the occuring distances empirically, so that the median of the occuring pairwise distances in the databse for a particular feature would be around 1. $\text{dist}_d(p, q)$ denotes this normalised distance between images p and q under descriptor d.

NN^k Browsing We use the same NN^k network structure as last year [4], [5]. Given a query image Q and a vector F of feature-specific similarities F_i, we compute the overall similarity between two images as the weighted sum over the feature specific similarities, i.e.

$$S(Q, I) = \sum_i w_i F_i$$

with $|\mathbf{w}| = \mathbf{1}$ and $0 \leq w_i \leq 1$. An image Q is then connected to an image I by a directed edge $I \rightarrow Q$ if and only if I is the nearest neighbour of Q for at least one combination of features, i.e. if there is at least one instantiation of the weight vector w such that it causes the image I to have the highest similarity $S(Q, I)$ among all images of the collection (excluding itself) (for more informations see [5].

3.4 Experiments

Manual runs: Two manual runs were carried out. One was the compulsory run with only the text feature permitted. For the other run, we used topic-specific weight vectors to weigh each of the eleven features described above. The weight vectors were our crude guesses of the optimal weight sets and were based on inspection of the query images only.

Interactive runs: Two interactive runs were carried out. In one run, the users were asked to use only the automated search functionality and without NN^k browsing. In a second run, users were invited to use both content-based search and NN^k browsing.

Block design With a total of four users, 24 topics and two system variants to be compared, we chose an experimental design that required each user to execute a total of 12 queries.

	T125-130	T131-136	T137-142	T143-148
U1	I	II		
U2		I	II	
U3			I	II
U4	II			I

where the rows and columns represent, respectively, the set of users and the set of topics.

3.5 Results

The results for the two interactive and the two manual runs are shown in Table 2.

System Variant	MAP
Interactive: Search	0.189
Interactive: Search and Browsing	0.200
TRECVID Interactive Median	0.154
TRECVID Interactive Max	0.423
Manual: Text only	0.051
Manual: Text + Visual Features	0.045
TRECVID Manual Median	0.053
TRECVID Manual Max	0.171

Table 2. Search task results for the interactive runs averaged over all 24 topics

4 Feature detection task

4.1 System

We attempted only one feature in the high-level feature extraction task - the “basketball” feature (No. 33). For training we selected 20 key frames judged relevant for the respective query of the last year’s search task. A simple nearest-neighbour approach was used for identifying relevant key frames in the test collection. Three visual features were chosen: Global HSV, Thumbnail and Convolution and for each similarity scores were computed as follows:

$$S(i) = \sum_{p \in P} \frac{1}{d_f(i, p)}$$

where $d_f(i, p)$ is the L_1 distance between the histograms of images i and p, and P is the set of all example images. Confidence values of the word “basketball” in LIMSI ASR transcripts were also used as an additional similarity score. For each feature the scores were normalised to have zero mean and unit variance, and the final similarity score was computed as their sum. The results were sorted by score and the top 2000 submitted as positive shots.

4.2 Results

System Variant	Avg prec
Imperial-01 (visual features and text)	0.1770
Imperial-02 (visual features only)	0.1246
TRECVID Median	0.1940

Table 3. High level feature extraction task – average precision results and hits at depth 2000 for feature 33 (basketball)

Our results for the two runs on feature 33 are shown in Table 3. As expected, the run that incorporates text performs better, and performance can be viewed as satisfactory given the simplicity of the approach.

5 Story boundary detection

5.1 System

We based out story detection system on the techniques used by the ANSES system developed in our group. Full details of this can be found in [9]. The ANSES system is live, capturing the BBC news each evening. It segments the news programme into stories and presents a summary to the user. The stories are searchable. Anecdotally users find the performance to be very accurate.

In contrast to the text-based methods deployed in the topic detection and topic tracking research field [1], we work under the assumption that story boundaries tend to coincide with video shot boundaries, and that the problem of story segmentation can be reduced to that of merging shots.

The output of the shot boundary detector is the timing information for the start and end of the shot and a single key frame to represent the shot. The exception to this was the run using only speech recognition, where we took the timing of the text segments as the starting set of boundary timings.

We have designed two basic approaches to story boundary detection that work on top of the shot boundary detector: a text-based approach using the subtitles or speech recognition transcripts and an approach using anchorperson shot detection. The two core techniques are described below. The details of how we applied these to the TRECVID task are in Section 5.2.

Text based story boundary detection

Segment merging algorithm. To decide whether to merge text segments each text segment is compared to each of its 5 neighbours on either side. When the similarity of two segments is greater than an empirically defined threshold, the two segments are merged with each other, and with any other segments that lie temporally between them. The corresponding video shots are also then merged.

The following function returning a score for the similarity of two text segments was used:

$$\text{Similarity} = \sum_{m} \frac{w(m)}{d(m)},$$

where $w(m)$ is the weight of word in match m according to its type and $d(m)$ is the number of subtitle lines between the two words in the match. Words, when matched, will generate a different score according to their type, as shown in Table 4. A word may have more than one type and scores are accumulated accordingly — for example 'Beijing' is both a location and a noun, and would score $40 + 10 = 50$. The 'Other words' category includes all non-stopped words that do not fit another category.

Word type	Score
Organisation	60
Person	60
First Person	40
Location	40
Date	30
Noun	10
Other words	5

Table 4. Empirically determined word-type weightings

Anchorperson-based story boundary detection In this method we work with the additional assumption that each news story starts with a studio anchorperson. Based on this assumption, we detect anchorperson shots and use the corresponding shot start time as the story start time.

The basis for the anchorperson detector is a k-NN based image retrieval system. The system was trained using prvious year's TRECVID data. Separate training was done for CNN news and ABC news (since their studios differ in presentation) as follows. All anchorperson shots were manually marked in 10 randomly selected broadcasts from the development set for each channel, and then 30 positive examples randomly selected from each of these subsets. 100 negative examples were randomly selected for each channel from the complement of each positive subset.

The k-NN method returns a ranking value for each key frame in the test collection, and it was necessary to set a threshold such that all key frames ranked above it would be classified as anchorperson shots. From the 20 manually marked broadcasts (10 for each channel) it was determined that on average 2% of key frames were anchorperson shots, and the initial threshold was set such that the k-NN method would classify a similar proportion of key frames from the test set as anchorperson shots.

Finally, for each shot output from the shot boundary detection process, if that shot was an anchorperson shot, we merged with it all following shots, up until the start of the next anchorperson shot.

5.2 Experiments

The TRECVID task had three required submissions. These are detailed below together with the approaches we used. We also submitted runs using two additional methods. The run ids given correspond to those in the results Table 5.3.

1. Required: Video + Audio. Our system did not use audio so for this task we were limited to video. We submitted 3 anchor person runs using differing thresholds. Run ids are of the form 1-anc-0.37.
2. Required: Video + Audio + LIMSI ASR (no transcripts, etc.). The text merging approach was used based on the the LIMSI ASR data. A single run, id 2-limsi-0.95, was submitted.
3. Required: LIMSI ASR (just the ASR). For this run we could not use our shot boundaries as these had been obtained using the video. We therefore used the timings of the ASR segments and then merged them using the text approach. A single run, 3-lv-1, was submitted.
4. Additional: Shot boundary + closed caption: Generally the closed caption stream is more reliable than the speech recognition text. However, it is not always accurately aligned with the video. We used the ASR to align the closed captions with video frames and then used the text-merging approach. Three runs were submitted using different thresholds. Run ids are of the form, 4-cc-0.05.
5. Additional: Anchor person + closed caption: For this run we used the output of the anchorperson detector method, followed by text-merging using closed captions. A single run, 4-comb0.37-0.5, was submitted. The drawback of this method is that boundaries merged in the first stage can not be recovered in the second.

5.3 Results

The results of our runs are presented in Table 5.3. Overall the results were disappointing, below the median. The best results were from text-merging using the closed captions. The trade off between recall and precision can clearly be seen.

Our system had originally been designed to work with the BBC news. The high performance with this data may be due to adaptations to the way the BBC subtitles are transmitted. We intend to develop our system further using the TRECVID training data and adding in further features using audio and motion.

Run id	Recall	Precision	Condition
1-anc-0.37	0.424	0.173	1
1-anc-0.40	0.330	0.263	1
1-anc-0.43	0.243	0.397	1
4-cc-0.05	0.315	0.291	4
4-cc-0.1	0.412	0.245	4
4-cc-0.3	0.532	0.190	4
4-comb0.37-0.5	0.352	0.272	4
2-limsi-0.95	0.561	0.107	2
3-lv-1	0.709	0.161	3
TRECVID Median	0.539	0.407	

Table 5. Story boundary detection task – results summary

6 Conclusions

Our performance in the interactive search task has ceded ground compared to last year, both in absolute figures and in comparison with other groups, although it still lies significantly above the median. This is likely to be a result of the growing requirement for motion analysis, a functionality our system does not currently support. Including browsing functionality does have a beneficial effect on performance. Analysis of the performance differences at the topic level reveals that NN^k browsing appears to result in a dramatic increase in performance for relatively hard queries where low-level features are less informative (such as Henry Hyder (128), Saddam Hussein (133), Sam Donaldson (135), Netanjahu (137) and provides no extra benefits when there is a closer correspondence between image content and low-level features (e.g. Ice Hockey (130), Fire (147)).

The results for the manual runs are largely in accord with the results from last year and confirm our conclusion that there is no evidence that inclusion of content-based features significantly improves retrieval performance beyond that of text-based search given reasonably reliable text annotation.

Acknowledgements: This work was partially supported by the EPSRC, UK.

References

1. J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study final report. In *Proceedings of the Broadcast News Transcription and Understanding Workshop (Sponsored by DARPA)*, Feb. 1998.
2. C. C. Gotlieb and H. E. Kreyszig. Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics and Image Processing*, 51:70–86, 1990.
3. R. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
4. D. Heesch, M. Pickering, S. Rüger, and A. Yavlinsky. Video retrieval using search and browsing with key frames. In *Proceedings of TREC Video Retrieval Evaluation (TRECVID, Gaithersburg)*, 2003.
5. D. Heesch and S. Rüger. NN^k networks for content-based image retrieval. In *26th European Conference on Information Retrieval*, pages 253–266. Springer-Verlag, April 2004.
6. P. Howarth and S. Rüger. Evaluation of texture features for content-based image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 326–324. Springer-Verlag, July 2004.
7. B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.
8. B. S. Manjunath and J.-S. Ohm. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11:703–715, 2001.
9. M. Pickering. *Video Retrival and Summarisation.* PhD thesis, Imperial College London, 2004.
10. M. J. Pickering, D. Heesch, R. O'Callaghan, S. Rüger, and D. Bull. Video retrieval using global features in keyframes. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the Eleventh Text REtrieval Conference (TREC-11)*, 2003.
11. D. Pye, N. J. Hollinghurst, T. J. Mills, and K. R. Wood. Audio-visual segmentation for content-based retrieval. In *5th International Conference on Spoken Language Processing, Sydney, Australia*, Dec. 1998.
12. H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Trans on Systems, Man and Cybernetics*, 8(6):460–472, 1978.
13. K. Tieu and P. Viola. Boosting image retrieval. In *5th International Conference on Spoken Language Processing*, Dec. 2000.
14. D. Travis. *Effective Color Display.* Academic Press, San Diego, CA, 1991.
15. H. J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia Systems*, 1:10–28, 1993.