Video Indexing and Retrieval at UMD

Christian Wolf
Laboratoire RFV
INSA de Lyon
69621 Villeurbanne cedex, France
wolf@rfv.insa-lyon.fr

David Doermann
Language and Media Processing Lab.
University of Maryland
College Park, MD 20742-3275, USA
doermann@cfar.umd.edu

Mika Rautiainen
MediaTeam Oulu
University of Oulu
P.O.BOX 4500, Finland
mika.rautiainen@ee.oulu.fi

Abstract

Our team from the University of Maryland and INSA de Lyon participated in the feature extraction evaluation with overlay text features and in the search evaluation with a query retrieval and browsing system. For search we developed a weighted query mechanism by integrating 1) text (OCR and speech recognition) content using full text and n-grams through the MG system, 2) color correlogram indexing of image and video shots reported last year in TREC, and 3) ranked versions of the extracted binary features. A command line version of the interface allows users to formulate simple queries, store them and use weighted combinations of the simple queries to generate compound queries.

One novel component of our interactive approach is the ability for the users to formulate dynamic queries previously developed for database applications at Maryland. The interactive interface treats each video clip as visual object in a multi-dimensional space, and each "feature" of that clip is mapped to one dimension. The user can visualize any two dimensions by placing any two features on the horizontal and vertical axis with additional dimensions visualized by adding attributes to each object.

1 Introduction

For the search evaluation we ran experiments for both run types using two different tools: a command line based query tool which allows us to query a precompiled database using text and speech (keywords), color, and binary feature results and a data browsing tool which represents each shot as a point in a space defined by the available features.

This paper is outlined as follows: the text detector used for feature extraction has been described previously in [12]. The experiments specific to the TREC competition are described in Section 2. The features used for the search task and the techniques we developed to query them are given in Section 3. Experiments are presented in Section 4, and Section 5 provides conclusions and describes the future directions of our research.

2 The feature extraction task: detection of overlay text

Our team participated in the feature extraction subtask with an overlay text detector developed at INSA de Lyon [12]. Text is detected on a frame by frame basis, integrated into a single image per occurrence, enhanced and binarized before being passed to a commercial OCR. For the TREC competition we used OCR software from Scansoft, which does not perform as well on our data as the Finereader software we used in [12], but it does come with a programable API which makes it easier to integrate it into our system. This is imperative given the amount of data we needed during the competition.

In order to increase the precision of the detector, we developed a classifier which takes the OCR output from each detected text box and classifies it either as a positive text result or a false alarm.
Table 2 shows OCR output examples for text and non-text. At first glance it seems to be straightforward for a human to judge between "meaningful" text and junk strings. However, we did not want to use a dictionary based approach for two reasons:

Non text examples		Text examples	
a yen Pu s1c~	i ~~~t~ ..~f: a1 t.	TONY RIYERA	Art Direction
v\~~~	ad	EUGENE PODDANY	URANIUM
.~a~~ 1r.	~! ~ ~._'_‘	EMERY NAWK~N5	92
~-	,(1f~rJ1	GERALD NEYIU	GzOTONS 92
.1	I~rs~-'.' i~u .r	D i recto r	arrtoms 142
j_ I	lsYf,7	CARL URBAN	234

Table 1: Examples for text and non-text OCR output.

- Screen text often contains words which cannot be found in a dictionary, such as names and identifiers containing digits.
- A part of the OCR output is very noisy (e.g. “arrtoms” instead of “atoms” in the examples).

We therefore used simple features based on the type and distribution of the characters in the string. We manually classified the set of possible characters into the 4 different groups “upper case”, “lower case”, “digits”, “non-text”. The input to the classifier is a feature vector containing the following two features:

$$F_1 = \frac{\text{Number of good characters (upper+lower+digits)}}{\text{Number of characters}} \quad F_2 = \frac{\text{Number of group changes}}{\text{Number of characters}}$$

We trained a linear classifier with Fisher’s linear discriminant [1] on a set of training strings and ranked all text instances by the classifier output.

3 Search: features and query techniques

For the second task, general search, we used all of the donated features (i.e. the results of the automatic speech recognition (ASR), the binary features, as well as the results of the automatic text recognition (ATR) described in Section 2 and the corresponding transcripts, and the temporal color correlograms already used during TREC 2001 and described in [9]).

In the following subsections we describe the different features and the query techniques necessary to use them in a query system. Finally, Sections 3.4 and 3.5 describe the two tools employed for our experiments.

3.1 Recognized text and speech

Recognized video text and speech transcriptions are treated in the same way by our system. For retrieval we used the Managing Gigabytes system[11] which is freely available[1]. This software is able to retrieve “documents” from one or more “collections”, defined by the user. In our case, documents were the text content obtained for each shot, with the three collections corresponding to different feature sources: Text, Speech and Text+Speech. Managing Gigabytes (MG) allows users to choose between two different query metrics: boolean queries and ranked queries, where the relevance of each shot is determined by the cosine similarity[11]. We allowed users to choose between the boolean query type and the ranked query type, with ranked queries being the default.

MG stems all document and query terms, so for example, looking for the term “produced” will also retrieve shots containing the term “producer” or “producing”. Unfortunately this feature cannot be turned off, so we could not measure it’s influence on the query results in the case of noisy transcripts.

[1] http://www.cs.mu.oz.au/mg

The MG system was developed for large collections of noise free documents, and therefore all hits are computed as exact matches (apart from the stemming technique). The ASR transcripts we received from the feature donors had been post processed with a dictionary and therefore contained only noise free text. The ATR transcripts on the other hand were quite noisy. For example the string "`Nick Chandler`"[2] had been recognized by the OCR in the shot as "`ni ck l6 tia ndler`", so a search for the keyword "chandler" would not retrieve the shot. We included a possibility of performing inexact matches using N-grams. The N-Gram of a string is the set of all sub strings of length $\geq N$. By performing a boolean query and OR-connecting all the substrings, exact matches of these substrings will be retrieved. The minimum length N of the substrings determines the level of "exactness". Lower values of N will retrieve documents with noisier keywords but also more irrelevant documents. We set the N empirically to 5. Applied to the example given above, an inexact query on the keyword "chandler" would result in the following boolean query:

`chand|handl|andle|ndler|chandl|handle|andler|chandle|handler|chandler`

which retrieves the desired shot but also a shot containing the transcript "`colleges cattlemen handlers of livestock`", clearly a false alarm.

3.2 Binary features

We used all of the donated binary features except CMU's face detector and MSRA's overlay text detector. These two detectors do not provide confidence information, which is needed for our feature combination technique.

Combining the features can be viewed as the classical problem of combining the results of multiple classifiers which has already received considerate attention in the pattern recognition and in the retrieval community [5, 3, 2, 6]. Training the combining classifier has been reported to considerately improve the performance of the total system in some cases, especially if the base classifiers use different features[2]. In our case, however, training data is not available since we are not allowed to use features from TREC's search test collection. Therefore, using fixed rules to combine the classifiers is the only available option.

In the following discussion, $C_{ij}(x)$ denotes the output of classifier j for class i, given the measurement vector x and $Q_i(x)$ the output of the combining classifier. We did not formulate the rules in a probabilistic way, since we cannot know if the output of the classifiers quantifies the posterior probabilities of the class memberships. The two most well known fixed combination rules are the product rule

$$Q_i(x) = \prod_j C_{ij}(x)$$

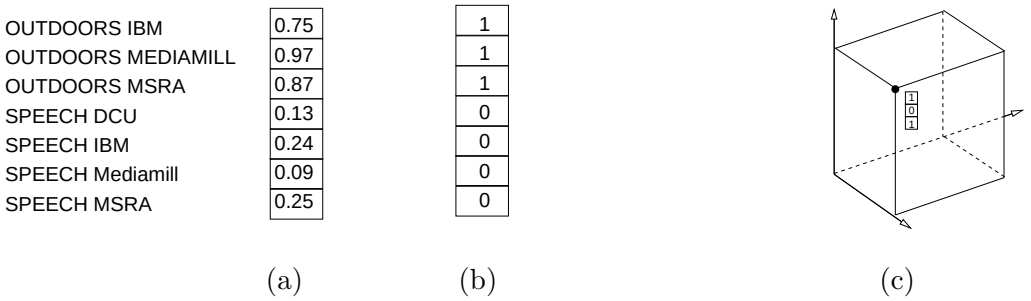
and the sum rule

$$Q_i(x) = \sum_j C_{ij}(x)$$

[5, 2]. If the feature representations are statistically independent and the classifiers produce posterior probabilities, then the product rule will produce an optimal classifier in the sense that it quantifies the true likelihood of the measurements given class k. Unfortunately, this is rarely the case [5]. The sum rule is considered to work well in cases where base classifiers have independent noise behavior as errors in one classifier are averaged out by the other classifiers.

We chose the more robust sum rule for this task, since we do not have enough information on the type of classifiers used for the base classifiers and how they have been trained. If one of the classifiers is

[2]The person "Nick Chandler" is not related to the "James H. Chandler" from TREC search topic 75.



(a) (b) (c)

Figure 1: An example shot vector (a) and an example query vector (b) for the query "*outdoors but not speech*". The feature space for a query on 3 features (c). The query vector is a corner point in the hyper cube.

weakly trained, then the product rule will create a combining classifier with very poor results, whereas the sum rule tends to reduce the effect of a poor base classifier. Since we do not want to classify the shot by the combined classifier alone, but are also interested in the output value of the classifier, we divide the output by the number of classifiers, which gives us the average of the classifier outputs. Combining the features this way, it is possible to create filters on the binary features, retrieving only shots with a certain confidence of containing a feature.

On the other hand, a user might prefer to submit a ranked query returning a set of shots "closest to " the desired features. In this case it is more convenient to use a vector space model for the set of classifiers since vector space distance functions can be used to order the shots. The actual space depends on the query specified by the user. The user may wish to retrieve shots with constraints on certain features but may also want to ignore others. As an example, in order to retrieve people walking on a beach, a user might want to return shots containing high confidences on the features *outdoors* and *people* and low confidences on the features *indoors* and *cityscape*, but is likely to ignore the feature *speech*. In our vector space model, each dimension in the vector space corresponds to a donated feature, which is constrained in the query, so the vector space is a sub space of the space spanned by all features. If, for example, the query puts constraints on the features *outdoors* and *speech* and *outdoors* has been donated by 3 teams whereas *speech* has been donated by 4 teams, then the feature space has $N = 7$ dimensions (see Figure 3.2a). For each shot s, the confidence outputs can be placed into a N dimensional feature vector u_s.

The shot vectors are compared with a query vector q, whose elements of the are set to 1 if the corresponding feature is desired by the user and to 0 if the corresponding feature is not (see Figure 3.2b). The space of all possible shot vectors is therefore bounded by an N-dimensional hypercube and the query vector is one of the corners of the hypercube (see Figure 3.2c).

The confidence output of all feature detectors is already scaled to the interval $[0-1]$, but the distribution of the confidence values in the interval may be different for different detectors, as is the case for the two features for *Indoors* donated by IBM and MSRA. We therefore considered the Euclidean distance $D_E(q, u_s) = \{(q-u_s)^T(q-u_s)\}^{-1}$ as well as the Mahalanobis distance $D_M(q, u_s) = (q - u_s)^T \sum^{-1} (q - u_s)$, which scales the distances according to the covariances Σ of the data set. During the experiments, however, the Euclidean distance outperformed the Mahalanobis distance by far. This can be explained with the different distributions of the confidence values of the different detectors.

3.3 Temporal color features

We incorporated a capability for color motion search by the temporal color features developed by the University of Maryland in collaboration with the University of Oulu and presented in [9]. These features

model the statistical distribution of the colors in the shot, but, unlike histograms, they also capture the spatial and temporal distribution of the colors. A correlogram is defined as:

$$\gamma_{c_i,c_j}^{(d)} = \Pr_{p_1 \in I_{c_i}^n, p_2 \in I^n}(p_2 \in I_{c_j}^n \Big| |p_1 - p_2| = d)$$

where I^n are the frames of the shot, p_1 and p_2 are two pixels and c_i and c_j are two colors. In each bin, the feature stores the probability that given any pixel p_1 of color c_i, a pixel p_2 at distance d is of color c_j among the shots frames I^n. For the Video TREC evaluation, the features have been computed as auto correlograms, and therefore $c_i = c_j$, resulting in a two dimensional structure. The distance between two features is calculated using the L_1 norm.

3.4 Querying

A command line interface allows the users to submit queries on the features with the techniques described above. The query results can be stored in 30 different result frames, allowing combinations of the results using different schemes. The following operations are supported by the interface:

- Keyword based queries on text, speech or both; with or without n-grams; and boolean or ranked.
- Ranked color queries.
- Ranked queries on binary features and filters on binary features.
- Combination of query results (AND/OR) including weighted combinations of the ranking of both queries. Assigning each query i a weight α_i, the shots in the combined set are ordered by a measure[3] m_s, which can be calculated from each shot s's ranking $r_{s,i}$ in the query results: $m_s = \sum_i \alpha_i \left(1 - \frac{r_{s,i}-1}{N+1}\right)$ where N is the total amount of shots in the combined set.
- Inspection of the keyframes of queries (thumbnails and full sized image).

3.5 Browsing

In addition to the command line query interface, the data set can be viewed graphically in a browsing tool developed at the University of Maryland. The user has the choice to either viewing the complete data set or to select a set of query results and to view only shots which have been returned by one or more of these queries. For each shot, the following features are available:

- The confidence of the binary features (one value per feature type).
- The text and speech recognition output and it's confidence values.
- The rank of the shot. The ranking information is scaled and normalized to make it intuitively conform to the confidence values of the binary features ("*higher means better*"): $m_s = 1 - \frac{r_{s,i}-1}{N+1}$
- A OR-combined rank-like measure computed on the ranking of the shot in all chosen queries. The weights α_i are chosen by the user when the queries are exported to the browsing tool.

Browsing allows users to view the shots as visual objects on a dynamic two-dimensional grid. The available features can be used in multiple ways to visualize the data set. The user can visualize the shots in two dimensions by placing any two features on the horizontal and vertical axis. Additional dimensions can be visualized by adding attributes to each object. Color, for example, can be used to represent a third feature dimension. Dynamic range sliders are provided for all features (not just the visible ones) so the user can dynamically modify the visibility of each feature and navigate through the collection. Attributes of each object can be configured and displayed as "tips" when the mouse passes near them.

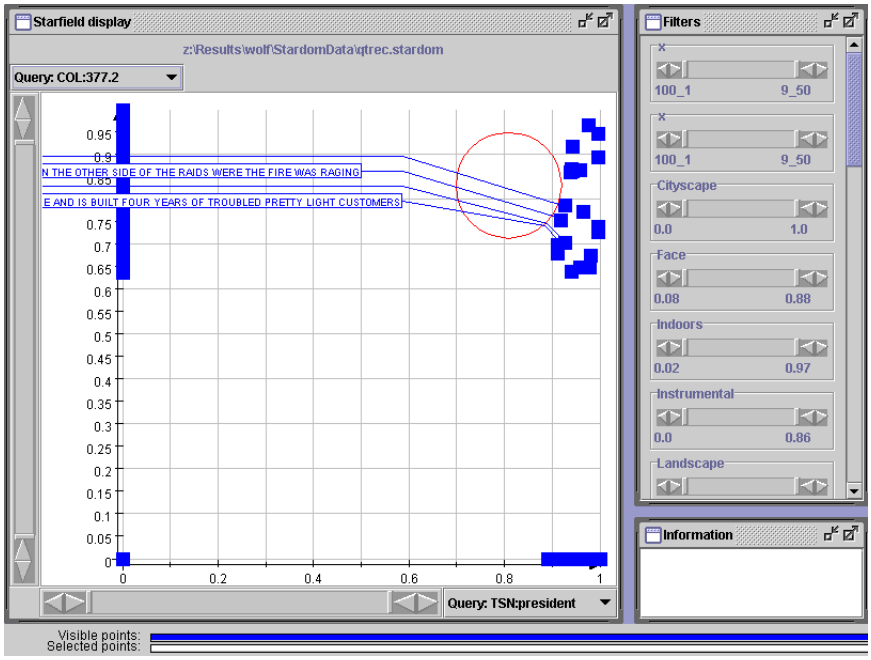


Figure 2: The graphical browsing tool "stardom".

Additionally, by clicking on a visual object or choosing a set of objects, their keyframes can be displayed and a media player can be launched to view the shot.

Figure 3.5 shows an example for a browsing session. The rank of a color query has been assigned to the y-axis and the results of a text/speech query (keyword: "president") has been assigned to the x-axis. All shots inside or touching the red circle are listed in the blue boxed list, for each shot the text/speech recognition results are displayed.

4 Experimental Results

In accordance with the TREC task definitions, we submitted 3 different runs:

Manual run The user adjusts the query parameters in the command line query tool to manually run queries and combines them to a single query. Our users where asked to write down the manual queries on a sheet of paper before actually performing them with the tool in order to avoid potential feedback from the subqueries.

Manual run/no ASR The same queries as above, but without speech recognition.

Interactive run The user is free to look at the results of each intermediate query returned by the command line tool and to adjust parameters and re-run them. Additionally, the query results can be viewed with the browsing tool for a closer inspection of the result sets and the mappings into the feature space. Based on these observations, queries can be re-run in the query tool. Hence, submitted results were queries run and combined in the query tool.

Four different users participated at the experiments. The users had been given dummy topics in order be able to get used to the system and the features. They did not know the real topics before they started working on them.

[3]The measure is not real ranking information since a value may occur several times in the combined set.

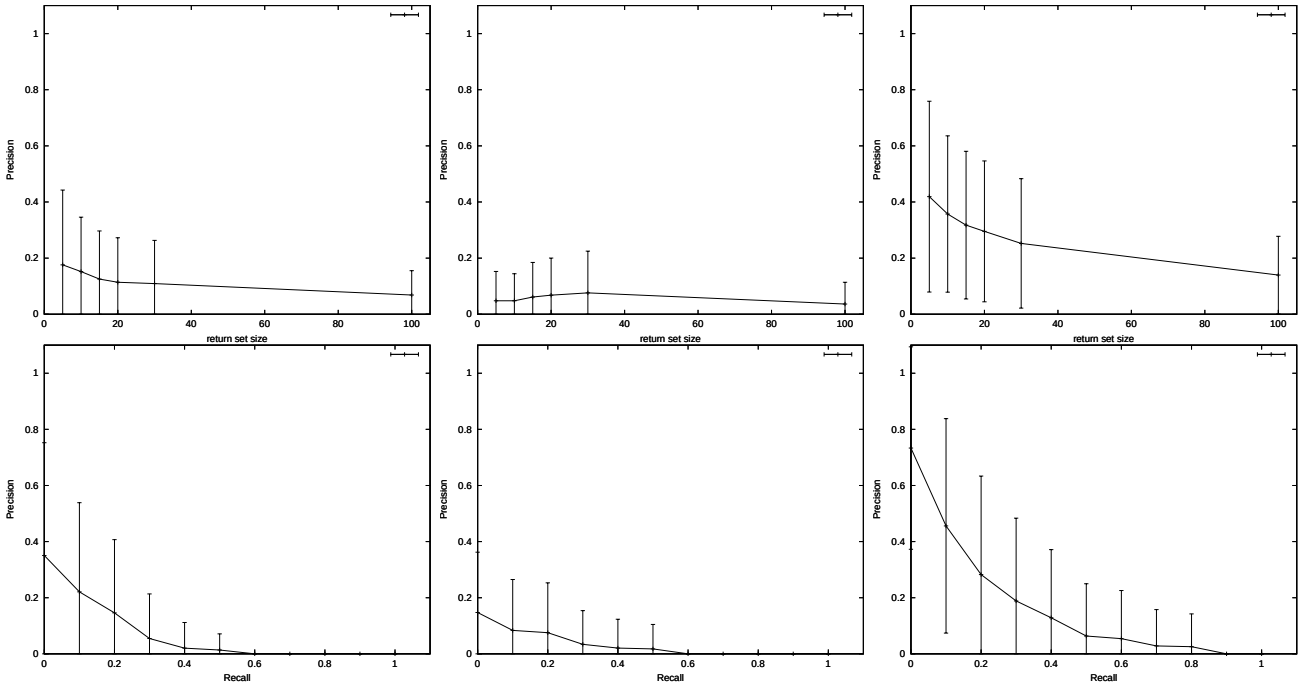


Figure 3: Precision at different result sizes averaged across the topics (top), and precision at different levels of recall averaged across the topics (bottom). From left to right: the manual run, the manual run without speech recognition and the interactive run

4.1 Search results

Figure 3 shows the precision curves vs. result size and recall, respectively, in graphical form. Recently, a new performance graph for the evaluation of content based image retrieval systems has been proposed[4]. It's advantage is the inclusion of generality information, i.e. information about the clustering of the relevant information in the database. This is essential if different systems operating on different databases are compared. Unfortunately, the measure cannot be computed for the TREC experiments, because it requires access to the precision at arbitrary sizes of the result set, information which is currently provided by NIST. Since all teams used the same database, it is possible to compare the different methods with the classic precision/recall graphs in order to determine which method is the best. However, a quantitative evaluation of the differences between the methods also makes the inclusion of the database generality necessary.

The interactive system performed best for query topics *76-James Chandler* (54% precision at 80% recall) and *97-living cells* because of the color features, and for *89-Butterfly* because of the color features and the ASR transcripts. Topics with medium sucess were *80-Musicians*, where a combination of color and ASR queries was used, and *77-George Washington*, *78-Abraham Lincoln*, *90-mountain* and *99-rockets*, which had been tackled with ASR only. For these topics, users performed queries with different keywords which were more or less related to the topic, and filterted the output with the binary features. For topic *86-Cities* a recall of 20% was achieved by a sole usage of color features. The user fed the results of the color queries back to new color queries and combined these results.

During the experiments we had signficant difficulty with the topics *79-Beach*, *87-Oil fields*, *88-Map*, *96-flag*, *98-locomotive approaching*. For four topics our experimenters could not find a single relevant shot: *75-Eddie Rickenbacker*, *81-Football players*, *85-Square arch*, *91-Parrots*. These are cases where the ASR transcripts did not return any relevant shots and the color features were not useful. Searching for

Figure 4: The results of a query on the speech transcripts using the keywords "rockets missile".

Eddie Rickenbacker, even the usage of the context taken from the request statement (e.g. the knowledge that he is the CEO of an airline) did not help to find a shot. The color features were particularly powerless in the topics *football*, *Beach* and *locomotive*, which are specified where a connection between the semantic concept and colors is very hard. A feature detecting drawings and cartoons could have been useful for the map topic. In some cases the users missed a possibility to explicitly specify colors or color ranges (e.g. for the topics *flag* and *parrots*).

4.2 The impact of ASR

Naturally, as the speech recognition transcripts are very close to the semantic meaning of the video, they have a significant impact on the search results. Nevertheless, the quality of keyword based queries on the transcripts highly depends on the topic. In general, the result sets of speech queries are very heterogeneous and need to be filtered, for example, by the results of binary filters.

Consider for example the results on the keywords "rocket, missile", shown in Figure 4. Adding additional keywords (e.g. "taking off") could increase the chances of retrieving rockets actually taking off as opposed to rockets on desks etc., but it also "thins" the semantic meaning of the query, in this case decreasing the probability to find shots about rockets or missiles in the result set. Another possibility is filtering the results of speech queries by the results of binary queries, a method applied frequently during our experiments.

4.3 The impact of ATR

The TREC 2002 video data set is a very bad collection to test the usability of recognized text for video retrieval. Text detection and recognition algorithms have been developed with different video material in mind such as newscasts, television journals and commercials, where text occurs much more frequently. In newscasts, the frequent appearance of names (interviewed people etc.) and locations rendered in overlay text makes indexing very powerful, especially in combination with face detection.

4.4 The impact of Color

As expected, the color features have been very useful in cases where the query shots where very different from other shots in terms of low level features, or where the relevant shots in the database share common color properties with the example query. For example, if they have been shot in the same environment.

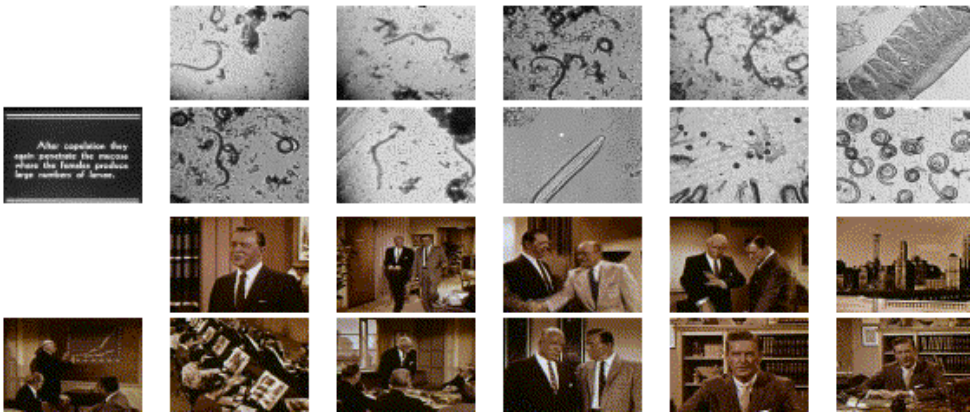


Figure 5: The results of color queries using example shots: searching for living cells under a microscope (a) searching for shots showing James H. Chandler (b).

text/speech	*swimming*	2	OR	10000	OR
text/speech	*shore*	1			
Landscape ≥ 0.3 Cityscape ≤ 0.5 Outdoors ≥ 0.5				1	
text/speech	*water*			2	
People≤ 0.5 Outdoors ≥ 0.5 Cityscape≤ 0.05 Indoors ≤ 0.75 Landscape≥ 0.5				1	

Table 2: The submitted query formulations for the interactive run of topic 79 *"People spending leisure time on the beach"*

Figure 5a, shows the result set of a color query on an example shot showing living cells under a microscope as requested for TREC search topic 97.

As another example of success, Figure 5b shows the results on the color queries on example shots showing James H. Chandler. Due to the special brownish color of the videos showing Mr. Chandler, the queries were very successful in retrieving shots from the same set of videos. In general, however, the color features need to be used very carefully, since no semantic meaning whatsoever is attached to them.

4.5 Discussion of the experiments

As stated above, the usefulness and impact of the different queries highly depends on the topics and on the example images and videos. In general, the speech recognition transcripts proved to be very important, although queries on them produced very heterogeneous outputs. The color features were only useful in isolated cases, where the targeted shots had very similar low level characteristics to the query images or videos. As expected, the binary features were very useful for filtering queries on the other features but not very useful as only features. Furthermore, the different donated features for the same feature type were not very correlated and tended to have a high amount of noise.

Table 4.5 shows an example of an interactive query submitted. The user realized, that the color features are not very useful in this case and concentrated on queries on the text and speech transcripts using different and more diverse keywords, filtered by binary features whose boundaries have been found also using the graphical browsing tool.

5 Conclusion and future work

A major emphasis in our work was put on the user and the experimental part of the work. We presented a query tool retrieving shots from user requests and a graphical tool which permits to browse through the feature and query space. Our future research will include:

- Exploiting the temporal continuities between the frames, e.g. as already proposed by the Dutch team during TREC 2001[10]. This seems to be even more important using overlay OCR results, since text is often rendered on neighboring shots.
- Combining the binary features (normalization, robust outlier detection etc.).
- Improving the graphical browsing interface. As our experiments showed, quick access to the keyframes and to the movie itself is a keystone in the success of the browsing tool. A browser showing tiny and enlargeable keyframes in the viewing grid instead of video objects should remarkably improve the performance of interactive queries.
- Adding additional features: e.g. explicit color filters, query by sketched example, motion, etc.

The most promising research in video retrieval points into the direction of the use of different features in order to attack the semantic gap from a materialistic point of view. Techniques combining different sources seem to be very successful [7, 3, 8]. However, sophisticated methods combining the large amounts of features are still missing and more theoretical research needs to be done in this area.

References

[1] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[2] R. Duin. The combining classifier: to train or not to train? In IEEE Computer Society, editor, *Proceedings of the ICPR 2002*, volume 2, pages 765–770, August 2002.

[3] G.E. Hinton. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, volume 1, pages 1–6, 1999.

[4] N. Huijsmans and N. Sebe. Extended Performance Graphs for Cluster Retrieval. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 26–32, 2001.

[5] J. Kittler, M. Hatef, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

[6] C.C. LuVogt. Learning to listen to the right information retrieval system. *Pattern Analysis and Applications*, (5):145–153, 2002.

[7] K. Messer, W. Christmas, and J. Kittler. Automatic sports classification. In IEEE Computer Society, editor, *Proceedings of the ICPR 2002*, volume 2, pages 1005–1008, August 2002.

[8] M.R.Naphade and T.S.Huang. Semantic Video Indexing using a probabilistic framework. In *Proceedings of the ICPR 2000*, pages 83–88, 3 September 2000.

[9] M. Rautiainen and D. Doermann. Temporal color correlograms in video retrieval. In IEEE Computer Society, editor, *Proceedings of the ICPR 2002*, volume 1, pages 267–270, August 2002.

[10] The Lowlands Team. Lazy users and automatic video retrieval tools in (the) lowlands. In NIST, editor, *The Tenth Text REtrieval Conference, TREC 2001*, pages 159–168, 2001.

[11] I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes - Compressing and Indexing Documents and Images*. Academic Press, San Diego, 2nd edition, 1999.

[12] C. Wolf, J.-M. Jolion, and F. Chassaing. Text localization, enhancement and binarization in multimedia documents. In *Proceedings of the ICPR 2002*, volume 2, pages 1037–1040, August 2002.