Interactive Experiments in Object-Based Retrieval

Sorin Sav, Gareth J.F. Jones, Hyowon Lee,
Noel E. O'Connor and Alan F. Smeaton

Adaptive Information Cluster & Centre for Digital Video Processing
Dublin City University, Glasnevin, Dublin 9, Ireland
sorinsav@eeng.dcu.ie

Abstract. Object-based retrieval is a modality for video retrieval based on segmenting objects from video and allowing end-users to use these objects as part of querying. In this paper we describe an empirical TRECVid-like evaluation of object-based search, and compare it with a standard image-based search into an interactive experiment with 24 search topics and 16 users each performing 12 search tasks on 50 hours of rushes video. This experiment attempts to measure the impact of object-based search on a corpus of video where textual annotation is not available.

1 Introduction

The main hurdles to greater use of objects in video retrieval are the overhead of object segmentation on large amounts of video and the issue of whether objects can actually be used efficiently for multimedia retrieval. Despite much focus and attention, fully automatic object segmentation is far from completely solved. Despite this there are already some examples of work which supports retrieval based on video objects. The notion of using objects in video retrieval has been seen as desirable for some time e.g [1], but only very recently has technology started to allow even very basic object-location functions on video.

In previous work we developed a video retrieval and browsing system which allowed users to search using the text of closed captions, using the whole keyframe and using a a set of pre-defined video objects [2]. We evaluated our system on the content of several seasons of the Simpsons TV series in order observe 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, aspect best illustrated in the work of the Informedia group at ACM Multimedia 2004 [3].

Our hypothesis in this paper is that there are certain types of information need which lend themselves to expression as queries where objects form a central part of the query. We have developed and implemented a system which can support object-based matching of video shots by using a semi-automatic segmentation process described in [4]. In this paper we investigate how useful this technique is for for searching and browsing very unstructured video, specifically the TRECVid BBC 2005 rushes corpus [5].

Research related to object-based retrieval is described in [6] where a set of homogeneous regions are grouped into an ad-hoc "object" in order to retrieve similar objects on a content of animated cartoons. Similarly in [7] there is another proposal for locating arbitrary-shaped objects in video sequences. Although these are not true object-based video retrieval systems they demonstrate video retrieval based on groups of segmented regions and are functionally identical to video object retrieval. In another approach in [8] object segmentation is performed on the query keyframe and this object is then matched and highlighted against similar objects appearing in video shots. This approach compensates for changes in the appearance of an object due to various artifacts presented in the video. Work reported in [9] addresses a complex approach to motion representation and object tracking and retrieval without actually segmenting the semantic object. Similar work, operating on video rather than video keyframes, is reported in [10] where video frames are automatically segmented into regions based on colour and texture, and then the largest of these is tracked through a video sequence.

The remainder of this paper is organised as follows. In the next section we outline the architecture of our object-based video retrieval system and briefly introduce its functionality. In section 3 we present the evaluation of object-based search functionality in an interactive search experiment on a test corpus of rushes video. The results derived from this evaluation are described and discussed in Section 4. Section 5 completes the paper summarising the conclusions of this study.

2 System description

In this section we outline the architecture of our object-based video retrieval system. 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 [11]. From the 50 hours of BBC rushes video footage we detected 8,717 shots, or 174 keyframes per hour, much less than for post-produced video such as broadcast TV news. For each shot we extracted a single keyframe by examining the whole shot for levels of visual activity using features extracted directly from the video bitstream. Rushes video is raw video footage which is unedited and contains lots of redundancy, overlap and wasted material in which shots are generally much longer than in post-produced video. The regular approach of choosing the first, last or middle frames as the keyframe within a shot would be quite inappropriate given the amount of "dead" time that is in shots within rushes video. Thus an approach to keyframe selection based on choosing the frame where the greatest amount of action is happening seems reasonable, although this is not always true and is certainly a topic for further investigation.

Each of the 8,717 keyframes was then examined to determine if there was at least one significant object present in the frame. For such keyframes one or more objects were semi-automatically segmented from the background using a segmentation tool we had developed and used previously [12]. This is based on performing an RSST-based [13] homogeneous colour segmentation. A user then scribbles on-screen using a mouse to indicate the region inside, and the 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 since not all keyframes contained objects.

Once the segmentation process is completed, we proceed to extract visual features from keyframes making use of several MPEG-7 XM [14] visual descriptors. These descriptors have been implemented as part of the aceToolbox [15] image analysis toolkit developed as part of the aceMedia project [16]. The descriptors used in our experiments were Dominant Colour, Texture Browsing and Shape compactness. The detailed presentation of these descriptors can be found in [17]. We extracted dominant colour and texture browsing features for all keyframes and dominant colour, texture browsing, and shape compactness features for all segmented objects. This effectively resulted in two separate representations of each keyframe/shot. We then pre-computed two 8,717 x 8,717 matrices of keyframe similarities using colour and texture for the whole keyframe and three 1,210 x 1,210 matrices of similarities between those keyframes with segmented objects using colour, texture and shape.

For retrieval or browsing of this or any other video archive with little metadata to describe it, we cannot assume that the user knows anything about its content since it is not catalogued in the conventional sense. In order to kick-start a search 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 [18] 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 the keyframes and the user is allowed to semi-automatically segment one object in the external image if they wish.

When these seed images have been 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 and 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 querying and browsing can continue until the user is satisfied. The overall architecture of our system is shown as Figure 1.

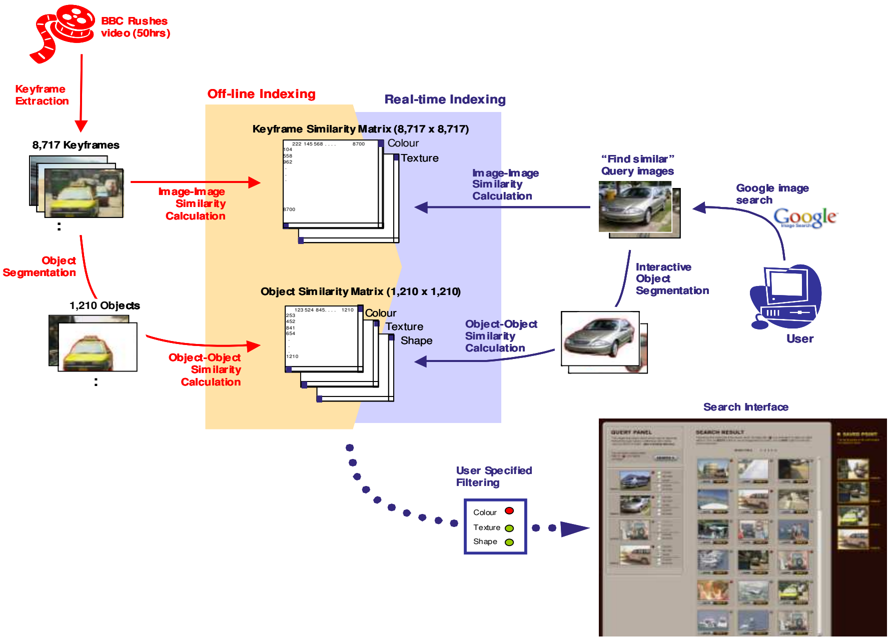

BBC Rushes video (50hrs)
Keyframe Extraction
Off-line Indexing
Real-time Indexing
Keyframe Similarity Matrix (8,717 x 8,717)
Colour
Texture
8,717 Keyframes
Image-Image Similarity Calculation
Image-Image Similarity Calculation
"Find similar" Query images
Google image search
Google
Interactive Object Segmentation
User
Object Segmentation
1,210 Objects
Object Similarity Matrix (1,210 x 1,210)
Colour
Texture
Shape
Object-Object Similarity Calculation
Object-Object Similarity Calculation
Search Interface
User Specified Filtering
Colour
Texture
Shape

Figure 1. System architecture overview

3 Experiments

In our experiments we aimed to evaluate how real users make use of object-based search functionality in contrast with image-based search. For this purpose we asked the users to perform a set of video searches with our system by using two identical looking search interfaces (like the one depicted at the bottom right in Figure 1). In one interface allowing object-based search users could use a combination of object and image searching, whereas in the other interface they were restricted (by disabling the object functionality) to using only whole image searching. For the reminder of this paper we refer to these interfaces as object-based interface and respectively image-based interface. The task given to the users is to find as many relevant shots for each predefined topic as possible. The effectiveness of a search interface being regarded as proportional with the number of relevant shots retrieved with that interface.

Each user was asked to perform a set of 6 separate search tasks with the object-based interface and a different set of 6 search tasks using the image-based interface. The users selected seed images from the Google image search and could semi-automatically segment objects in these images if they considered that useful for their search. The segmentation step is not performed when using the image interface. The users were instructed to save all relevant shots retrieved. At any stage during the search the user can add or remove images from the query either from the retrieved images or from the external resource.

We allocated only a 5 minute period for task completion for each of the 12 searches completed by each user. The objective of the time limit is was to put participants under pressure to complete the task within the available time. Users were offered the chance to take a break at session's half-time should they feel fatigued.

3.1 Search Topics Formulation

As described earlier in the paper, running shot boundary detection on the rushes corpus returned 8,717 shots with one keyframe per shot. 1,200 representative objects were selected and subsequently extracted from these keyframes.

For this experiment we required a set of realistic search topics. We based our formulation of the search topics on a set of over 1,000 real queries performed by professional TV editors at RTÉ, the Irish national broadcaster's video archive. These queries had previously been collected for another research project. The BBC rushes corpus consists of video recorded for a holiday program. One member of our team played through all the video and then eliminated queries which we knew could not be answered from the rushes collection. We then removed duplicate queries and similar, subsumed or narrow topics, ending with a set of 26 topics for which it is likely to find a reasonable number of relevant shots within this collection. Of these, 24 topics where used as search tasks and the other 2 as training during our users' familiarisation with the system. In the selection of search topics we did not consider whether they would be favorably inclined towards a particular search modality (object-based or image-based).

3.2 Experimental Design Methodology

In our experimental investigation we followed the guidelines for design of user experiments recommended by TRECVid [5]. These guidelines were developed in order to minimise the effect of user variability and possible noise in the experimental procedure. The guidelines outline the experimental process to be followed when measuring and comparing the effectiveness of two system variants (object/image based search versus image-only based search) using 24 topics and either 8, 16 or 24 searchers, each of whom searches 12 topics. The distribution of searchers against topics assumes a Latin-square configuration where a searcher performs a given topic only once and completes all work on one system variant before beginning any work on the other variant.

We chose to run the evaluation with 24 search topics and 16 users, with each user searching for 12 topics, 6 with the object/image based search and another 6 with the image-only based search. Our users were 16 postgraduate students and postdoctoral researchers: 8 people from within our research group with some prior exposure to video search interfaces and video retrieval experiments and another 8 people from other research fields with no exposure to video retrieval. Topics were assigned randomly to searchers. This design allows the estimation of the difference in performance between the two system variants free from the main (additive) effects of searcher and topic.

3.3 Experimental Procedure

In order to accommodate the schedules of users we ran experimental sessions with 4 users at a time. The search interface and segmentation tool were demonstrated to the users and 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 working with the retrieval system. Following these, the main search tasks began.

Users were handed a written description of the search topics. The topics were introduced one at a time at the beginning of each search task such that users would not be exposed to the next search topic in advance. This was done in order to reduce the influence that the current query and retrieved shots may have in revealing clues for the subsequent search topics. As previously stated, users were given 5 minutes for each topic and were offered the chance to take a break after completing 6 search topics. At the end of the two sessions (object/image and image-only based searching), each user was asked to complete a post-experiment questionnaire.

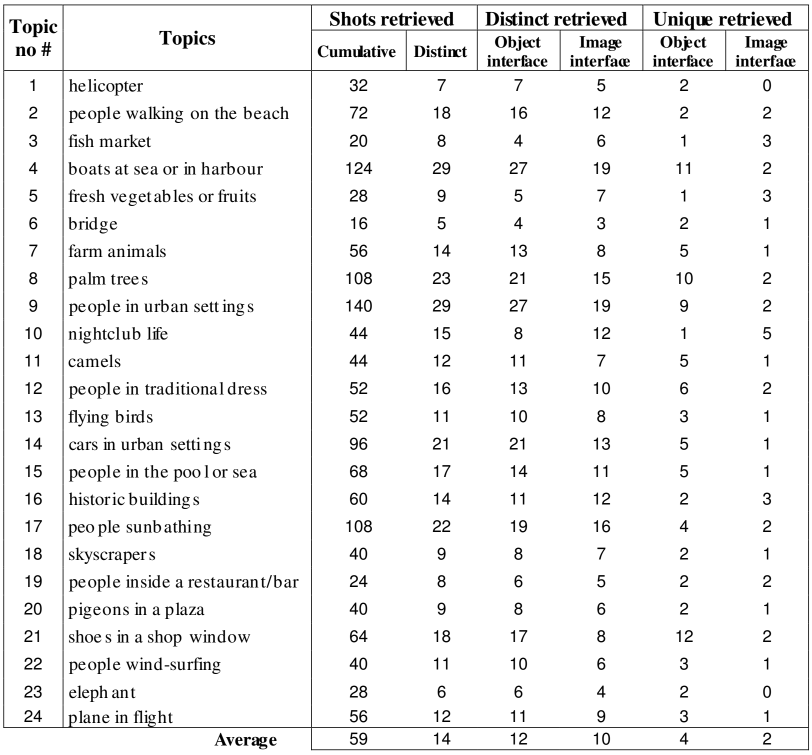
Each individual's interactions were logged by the system and one member of our team was present for the duration of each of the sessions to answer questions or handle any unexpected system issues. The results of users' searching (i.e. saved shots) were collected and formed the ground-truth for evaluation. The rationale behind doing this is that the shots saved by a user are assumed to be relevant and in terms of retrieval effectiveness for each system what we measure is how many shots, all assumed to be relevant, have users managed to locate and to explicitly save as relevant.

4 Results derived from experiments

For each topic we have collected a time-stamp log of the composition of each search at each iteration. Additionally in order to complement the understanding of objective measures we collected subjective observations from users through post-experiment questionnaires.

4.1 Evaluation metrics

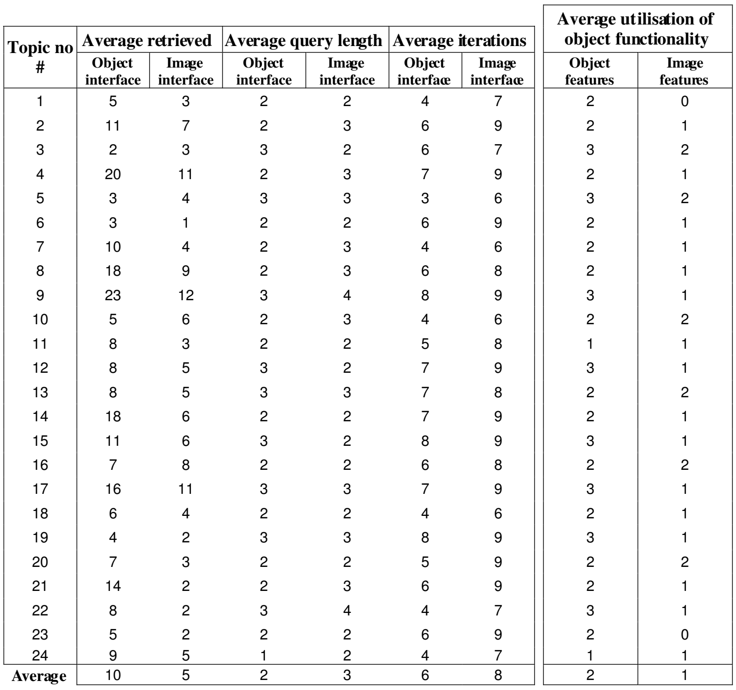
Since we did not have a manual relevance ground-truth for our topics, we assumed the shots saved by users during the interactive search to be relevant and used them as our recall baseline. Although we do not have any independent third party validation of the relevance of the saved shots our users were under instruction to only save shots they felt were relevant to the search topic, so this is not as unreasonable assumption. Naturally there may be other relevant shots in the collection, which were not retrieved by our users, but in the absence of exhaustive ground-truth we cannot know how many such shots are there. However our goal was to observe how real users make use of the object-based search functionality and that can be inferred even without an absolute ground-truth.



Topic no #	Topics	Shots retrieved		Distinct retrieved		Unique retrieved	
		Cumulative	Distinct	Object interface	Image interface	Object interface	Image interface
1	helicopter	32	7	7	5	2	0
2	people walking on the beach	72	18	16	12	2	2
3	fish market	20	8	4	6	1	3
4	boats at sea or in harbour	124	29	27	19	11	2
5	fresh vegetables or fruits	28	9	5	7	1	3
6	bridge	16	5	4	3	2	1
7	farm animals	56	14	13	8	5	1
8	palm trees	108	23	21	15	10	2
9	people in urban settings	140	29	27	19	9	2
10	nightclub life	44	15	8	12	1	5
11	camels	44	12	11	7	5	1
12	people in traditional dress	52	16	13	10	6	2
13	flying birds	52	11	10	8	3	1
14	cars in urban settings	96	21	21	13	5	1
15	people in the pool or sea	68	17	14	11	5	1
16	historic buildings	60	14	11	12	2	3
17	people sunbathing	108	22	19	16	4	2
18	skyscrapers	40	9	8	7	2	1
19	people inside a restaurant/bar	24	8	6	5	2	2
20	pigeons in a plaza	40	9	8	6	2	1
21	shoes in a shop window	64	18	17	8	12	2
22	people wind-surfing	40	11	10	6	3	1
23	elephant	28	6	6	4	2	0
24	plane in flight	56	12	11	9	3	1
	Average	59	14	12	10	4	2

Table 1. Size-bounded recall by search topic

From the logged data we derived the set of measures presented in Tables 1 and 2. The measures are shown for each search topic separately. The *shots retrieved* measure represents the total number of shots saved by all users for each search topic irrespective of the search interface used. The *cumulative* column gives the sum of shots saved by all users including the duplication of shots when saved by different users. The *distinct* value is obtained from the above cumulative number by removing duplicate shots. This value shows how many relevant shots were found for each topic. The *distinct retrieved* shots are then divided into shots saved with the object-based and with the image-based interface respectively. The *unique retrieved* value gives the number of distinct shots retrieved with only one of the search interfaces.



Topic no #	Average retrieved		Average query length		Average iterations		Average utilisation of object functionality	
	Object interface	Image interface	Object interface	Image interface	Object interfaœ	Image interfaœ	Object features	Image features
1	5	3	2	2	4	7	2	0
2	11	7	2	3	6	9	2	1
3	2	3	3	2	6	7	3	2
4	20	11	2	3	7	9	2	1
5	3	4	3	3	3	6	3	2
6	3	1	2	2	6	9	2	1
7	10	4	2	3	4	6	2	1
8	18	9	2	3	6	8	2	1
9	23	12	3	4	8	9	3	1
10	5	6	2	3	4	6	2	2
11	8	3	2	2	5	8	1	1
12	8	5	3	2	7	9	3	1
13	8	5	3	3	7	8	2	2
14	18	6	2	2	7	9	2	1
15	11	6	3	2	8	9	3	1
16	7	8	2	2	6	8	2	2
17	16	11	3	3	7	9	3	1
18	6	4	2	2	4	6	2	1
19	4	2	3	3	8	9	3	1
20	7	3	2	2	5	9	2	2
21	14	2	2	3	6	9	2	1
22	8	2	3	4	4	7	3	1
23	5	2	2	2	6	9	2	0
24	9	5	1	2	4	7	1	1
Average	10	5	2	3	6	8	2	1

Table 2. Average size-bounded recall by search topic

Table 2 shows the average values obtained during the 4 executions (by 4 users) of a search topic and each interface. All values are rounded to the nearest integer value. The *average retrieved* shots gives the mean number of distinct shots saved. The *average query length* shows how many images/objects have been used for each query, and *average iterations* presents the number of iteration runs for each search task. The last distinct column of this table measures the *average utilisation of object functionality* in terms of average number of images for which object features and/or global image features have been used within the object-based search interface.

4.2 Results interpretation

As shown by the *shots retrieved* values in Table 1 from the comparison between the *cumulative* and *distinct* values the sets of shots saved by different users largely overlap, which means that most users were able to find the same relevant shots although they may have used a different combination of query images or features. However during the experiments we observed that most users tended to initiate the search tasks from the same Google retrieved images, usually those found on the first page. Thus it is likely that most users have followed closely related search paths.

The number of *distinct retrieved* shots, given in Table 1 provides a measure of recall bound by the number of saved shots. By comparing the number of distinct shots retrieved with each search interface it can be observed that users found more relevant shots with the object-based interface. However that is not true for all search topics. For few search topics such as *fish market*, *bridge*, *nightclub life* and *historic building* searching on the image-based interface seemed to provide better results. These topics seem to be more suited to global image feature searching and although such features were also available on the object-based interface, users made only limited use of them, focusing mostly on object features. Additionally it is clear that except for the *bridge* topic, for the other three topics it is relatively difficult to define what images/objects will provide a good initial query. The object-based retrieval seems to provide not only better recall but also helps with locating shots that are not found by using image-only searching.

The average number of retrieved shots shows that object features provide better searching power than global features alone. The *average query length* and *average iterations* values are somehow correlated since performing an object-based search involves some time dedicated to segmenting objects which invariably reduces the time allocated to actually searching and therefore decreases the query length and the number of search iterations a user will be able to perform. The results shows that although using shorter queries and less iterations, object-based search compensates through the additional discerning capacity provided by the object's features. The *average utilisation of object functionality* shows that searchers have largely employed object-based features when available. This was confirmed as well by users' feedback provided in the post-experiment questionnaire.

5 Conclusions

In this paper we have described an empirical TRECVid-like evaluation of object-based video search functionality in an interactive search experiment. This was done in an attempt to isolate the impact of object-based search taking as an experimental collection the BBC rushes video corpus where text from automatic speech recognition (ASR), from video OCR, and from closed captions is not available. Sixteen users each completed 12 different searches, each in a controlled and measured environment with a 5 minutes time limit to complete each search.

The analysis of logged data corroborated with observations of user's behaviour during the search and with the feedback provided by users show that object-based searching consistently outperforms the image-based search. This result goes some way towards validating the approach of allowing users to select objects as a basis for searching video archives when the search dictates it as appropriate, though the technology to do this, is still under development for larger scale video collections.

6 Acknowledgments

BBC 2005 Rushes video is copyright for research purposes by the BBC through the TRECVid IR research collection. Part of this work was supported by Science Foundation Ireland under grant 03/IN.3/I361. We are grateful for the support of the *aceMedia* project which provided the aceToolbox image analysis toolkit.

References

1. E. Oomoto and K. Tanaka. Ovid: Design and implementation of a video-object database system. In IEEE Transactions on Knowledge and Data Engineering, vol 5, no.4, 1993.
2. A.F. Smeaton and P. Browne. A Usage Study of Retrieval Modalities for Video Shot Retrieval. Information Processing and Management (in press), 2006.
3. A. Hauptmann and M. Christel. Successful Approaches in the TREC Video Retrieval Evaluations. In Proceedings of ACM Multimedia, 2004.
4. S. Sav, H. Lee, A.F. Smeaton, N.E. O'Connor, and N. Murphy. Using Video Objects and Relevance Feedback in Video Retrieval. In Proceedings of the SPIE Conference on Multimedia Systems and Applications VIII, Boston, Mass., November 2005.
5. TRECVid Evaluation, available at http://www-nlpir.nist.gov/projects/trecvid
6. L. Hohl, F. Souvannavong, B. Merialdo, and B. Huet. Enhancing latent semantic analysis video object retrieval with structural information. In ICIP 2004 - International Conference on Image Processing, 2004.
7. B. Erol and F. Kossentini. Shape-based retrieval of video objects. In IEEE Transactions on Multimedia, vol 7, no.1, 2005.
8. J. Sivic, F. Shaffalitzky, and A. Zisserman. Efficient object retrieval from videos. In EUSIPCO 2004 - European Signal Processing Conference, 2004.
9. C.-B. Liu and N. Ahuja. Motion based retrieval of dynamic objects in videos. In Proceedings of ACM Multimedia, 2004.
10. M. Smith and A. Khotanzad. An object-based approach for digital video retrieval. In ITCC 2004 - International Conference on Information Technology: Coding and Computing, 2004.
11. P. Browne, C. Gurrin, H. Lee, K. McDonald, S. Sav, A.F. Smeaton, and J. Ye. Dublin City University Video Track Experiments for TREC 2001. In TREC 2001 - Proceedings of the Text REtrieval Conference, 2001.
12. T. Adamek and N.E. O'Connor. A Multiscale Representation Method for Nonrigid Shapes With a Single Closed Contour. In IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 5, May 2004.
13. E. Tuncel, L. Onural. Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2D affine motion modelling. In IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, no. 5, August 2000.
14. MPEG-7(xm) version 10.0, ISO/IEC/JTC1/SC29/WG11, N4062, 2001.
15. N.E. O'Connor, E. Cooke , H. LeBorgne , M. Blighe and T. Adamek. The AceToolbox: Low-Level Audiovisual Feature Extraction for Retrieval and Classification. In IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, London, UK, 2005.
16. The AceMedia project, available at http://www.acemedia.org
17. B. Manjunath, P. Salembier, and T. Sikora. Introduction to MEPG: Multimedia Content Description Standard. New York: Wiley, 2001.
18. The Google image search page, available at http://images.google.com