TREC-10 Experiments at University of Maryland CLIR and Video

Kareem Darwish,[1] David Doermann, Ryan Jones, Douglas Oard and Mika Rautiainen[2]
Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742

Abstract

The University of Maryland Researchers participated in both the Arabic-English Cross Language Information Retrieval (CLIR) and Video tracks of TREC-10. In the CLIR track, our goal was to explore effective monolingual Arabic IR techniques and effective query translation from English to Arabic for cross language IR. For the monolingual part, the use of the different index terms including words, stems, roots, and character n-grams were explored. For the English-Arabic CLIR, the use of MT, wordlist based translation, and non-dictionary words transliteration was explored. In the video track, we participated in the shot boundary detection, and known item search with the primary goals being to evaluate existing technology for shot detection and a new approach to extending simple visual image queries to video sequences. We present a general overview of the approaches, summarize the results in discuss how the algorithms are being extended.

1 CLIR Track

1.1 Introduction

For the CLIR track, we were interested in testing the effects of the choice of Arabic index terms, the use of morphology, and transliteration of words that are not in the dictionary. To test the effects before the ad-hoc TREC runs, we used a small Arabic collection called Zad. In the ad-hoc experiments we relied on insight gained from the small Arabic collection. In post-hoc TREC experiments, we examined the effects of different index terms on the Arabic monolingual and English to Arabic cross language retrieval results.

1.2 Methodology

Many new techniques were employed for ad-hoc TREC runs. The ideas were initially tested on a small side collection to verify the effectiveness of the proposed techniques. The collection is called Zad which was provided by Al-Areeb Electronic Publishers, LLC [2]. The collection contains 4,000 documents. The documents were extracted from writings of the thirteenth century scholar Ibn Al-Qayim and cover issues of history, jurisprudence, spirituality, and mannerisms. Also, there are 25 queries with their relevance judgments associated with the collection. The queries are typically 3-6 words long and are available in Arabic and English. The author developed the queries in Arabic, generated the relevance judgments by exhaustively examining the documents in the collection, and translated them to English.

The techniques addressed the choice of Arabic index terms, the use of morphology, and transliteration of words that are not in the dictionary.

To ease work in Arabic, Arabic letters were transliterated to English letters. Also, some letters normalizations were applied and all diacritics were removed. Table 1 shows the mappings between the Arabic letters and their transliterated representations.

---

[1] Authors are listed in alphabetical order

[2] Visiting from the Media Team, University of Oulu Finland

أ، ا، آ، إ،	A	ئ، ؤ، ء	A	ب	b
ت	t	ث	v	ج	j
ح	H	خ	x	د	d
ذ	O	ر	r	ز	z
س	s	ش	P	ص	S
ض	D	ط	T	ظ	Z
ع	E	غ	g	ف	f
ق	q	ك	k	ل	l
م	m	ن	n	ه	h
و	w	ي ،ى	y	ة	p

Table 1: English transliteration of Arabic characters

Notice that some letters such as {ى, ي} and {ا, آ, أ, إ, ء ، ؤ ،ئ} were normalized to "y" and "A" respectively. For the case of {ى, ي}, they are often used interchangeably for each other because of different orthographic conventions or common spelling errors. For the case of {ا, آ, أ, إ, ء ، ؤ ،ئ}, they represent different forms of the letter hamza.

As for a stop-word list, we used the list that is distributed with Sebawai which includes 130 particles and pronouns [6]. Finally we used the default settings of InQuery with stemming disabled and case sensitivity using the –nostem and –case switches respectively.

For query translation, we used an online machine translation (MT) system developed by Sakhr called Tarjim and a bilingual dictionary [9]. The dictionary was built by extracting unique terms from a 200-megabyte collection of news articles sending them to Tarjim for translation [8]. When our wordlist was sent to Tarjim, Tarjim produced single word translations of the words without regard for context.

1.2.1 Arabic index terms

Several papers were published comparing the use of words, stems, and roots as index terms for purposes of retrieval. All of the studies showed that stems outperformed words and roots outperformed stems [1][3]. We tested the claim using the Zad Arabic document collection. By testing on Zad, we noticed no statistical significance in mean average precision between words, stems, and roots. We thus tried using a combination of words and roots as index terms and the performance was significantly better than using any of them alone. This is a case when using a combination of evidence outperforms using any single evidence alone. For significance testing, we used a paired two-tailed *t*-test. If the p-value of the test was below 0.05, we assumed the difference to be significant.

We investigated other index terms which were character n-grams for both words and roots. We used a combination of character n-grams of different length. For words we used a combination of 3-5 grams and for roots we used 2-4 grams. In combining the n-grams, all the n-gram tuples replace the existing word. For example, the word "Arabic" would be replaced by {Ara, rab, abi, bic}, {Arab, rabi, abic}, and {Arabi, rabic}. Although character n-grams did not outperform words or roots, using the combination of words, roots, and character n-grams of words and roots together was significantly better than any pervious run. Table 2 summarizes the results of using different Arabic index terms on the side collection.

Index term	Mean Avg. Precision
Words	0.3939
Stems	0.4158
Roots	0.4486
Word & Roots	0.4979
Word character n-grams	0.4885
Word and root character n-grams	0.5717

Table 2: Summary of results on side collection of choosing different index terms.

1.2.2 Arabic morphology

Since previous research indicated that using roots as index terms improved mean average precision, two morphology engines capable of generating roots were compared. The first is ALPNET [4][5]. ALPNET has an inventory of 4,500 roots and for any given word, it generates possible roots in random order. The second is Sebawai, which was developed by the first author. Sebawai has an inventory of 10,500 roots and uses a heuristic that guesses which of the roots is most likely.

On the Zad collection, we conducted 4 experiments in which we examined indexing using roots only. The first two experiments involved indexing one root and two roots from ALPNET. For the other two, the experiments involved indexing using the top root and the top two roots from Sebawai. Using Sebawai's guess of the most likely root resulted in a higher mean average precision than when using one root from ALPNET. Further, using the first two roots from ALPNET slightly improved mean average precision, but the improvement was not statistically significant. Using the top two roots of Sebawai significantly harmed retrieval. A likely reason for the fall in mean average precision when the second root was introduced is that the second root amounted to noise. Table 3 summarizes the results of using roots from the two analyzers.

Index term	Mean Avg. Precision
ALPNET – 1 root	0.34
ALPNET – 2 root	0.36
Sebawai – 1 root	0.45
Sebawai – 1 root	0.29

Table 3: summary of results on side collection of using different morphological analyzers

1.2.3 Transliteration and matching of words that are not in the dictionary

For cross-language (CL) runs, we used an MT system in addition to a bilingual English to Arabic dictionary. Each English query was replaced with the full MT suggested translation and the word-by-word translation of query using the bilingual dictionary.

The MT system automatically transliterated words that did not appear in its internal dictionary. However, the suggested MT transliterations were often crude and incorrect. Detecting which words were in the MT lexicon and which ones were transliterations was beyond the scope of the work.

For the word-by-word dictionary based translation, we employed a transliteration technique for words that were not found in the dictionary. We assumed that the words that were not in the dictionary were mostly named entities and required transliteration. The goal of the transliteration technique is to find possible Arabic words that correspond to the given English word. The process involved transliteration, matching, and clustering.

Transliteration: All the English letters are mapped to the closest Arabic sounding letters. For example, the letter "r" is mapped to "ر". Letter combinations such as "ch" and "sh" are recognized and mapped to Arabic. Some letters such "j" and "g" are normalized to one letter. Table 4 lists the English to Arabic Transliteration mappings.

a*	A	b	b	c[iey]	s
c	k	d	d	[aeiou]	#**
f	f	g	g	h	h
j	g	k	k	l	l
m	m	n	n	p	b
q	q	r	r	s	s
t	t	v	f	x	k
y*	y	z*	z	th	O
al-	#	ala	A	[sc]h	P

Table 4: English to Arabic transliteration mappings (* initial letter(s) in the word, ** # represents nothing)

Matching: For the matching the transliterated words to the words in the collection to be searched, the prefixes {w,wAl,Al,wb,[wlbfk]} were removed from all the words; all the vowels are dropped; and some Arabic letters were normalized. Table 5 lists all the normalizations of Arabic letters.

[Ss]	s	[Zz]	z	[xk]	k	[AE]	A
[Hh]	h	[Tt]	t	[gj]	g	p	#

Table 5: Arabic letter normalizations (* # represents nothing)

Clustering: after the possible Arabic transliterations are found, all are used together in the Arabic queries using InQuery's #syn operator which sets all of them as synonyms to each other.

The effect of this technique is not completely clear given that most of the words in the queries for the side collection and TREC were in the bilingual dictionary.

1.3 Experiment Design

1.3.1 Arabic Monolingual Run

Automatic Arabic Run: Based on our experiments on Zad collection, we used words, stems, roots, character n-grams for words, and character n-grams for roots to index the TREC collection. To obtain stems and roots, we used the top suggestions of Sebawai. For n-grams, we used 2-4 character n-grams for roots and 3-5 character n-grams for words.

Manual Arabic Run: For the manual runs, the title, description, and narratives were used along with words that were manual introduced by the authors. We removed stop structures from queries such as " المقالات المتعلقة" (the articles relating to) and examples of what is not relevant. The final queries were run in exactly the same way as the automatic Arabic setup.

Post-hoc experiments: After the relevance judgments were available we explored the use of different index terms on our retrieval effectiveness. We examined indexing using words only, stems only, roots only, word character trigrams, root character bigrams, and words, stems, and roots together. The queries were automatically formulated using the full text of the title and descriptions of the queries.

1.3.2 English-Arabic CLIR Runs

For the CLIR runs, we tested three different configurations as follows:

Basic Configuration: All the queries were translated using Tarjim only. The output of the MT system was fed to the automatic Arabic IR configuration described above. It is noteworthy that Tarjim transliterates the words that do not appear its dictionary.

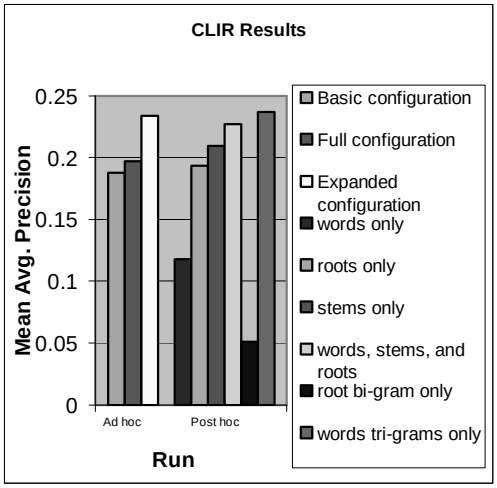
Full Configuration: In this configuration, the English queries were translated using Tarjim and the bilingual dictionary. If a word is not found in the dictionary, the word is transliterated, matched to Arabic words in the TREC collection, and the matches were clustered in the manner described above. The outputs of the MT system, the dictionary based translation, and the transliteration are combined and fed to the automatic Arabic IR configuration.

Expansion Configuration: The expansion configuration is identical to the full configuration but with expansion using blind relevance feedback on the English and the Arabic sides. For expansion on the English side, we used Associated Press articles from 1994-1998. They were part of the North American News Text Corpus (Supplement) and AP World Stream English Collection from the Linguistic Data Consortium [7]. The expansion collection was searched using the English queries without modification and the top 10 returned documents for every query were used to expand the query. For the expansion on the Arabic side, the TREC collection was used for expansion. The AFP collection was searched using the Arabic queries, which include the roots and n-grams, and the top 10 retrieved documents for ever query were used to expand the queries.

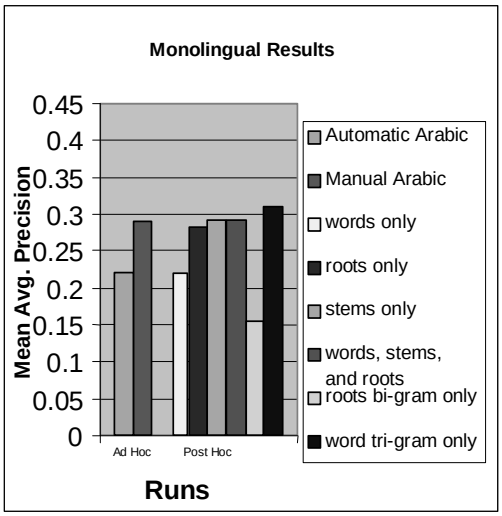
Post-hoc experiments: We examined indexing using words only, stems only, roots only, word character trigrams, root character bigrams, and words, stems, and roots together. The titles and descriptions of the queries were automatically translated using Tarjim alone.

1.4 Results

Official cross language runs (Ad hoc)	
Run	**Mean Avg. Precision**
bc - Basic configuration	0.19
fc - Full configuration	0.20
xp - Expanded configuration	0.23
Post hoc runs (all basic configuration)	
w - words only	0.12
r - roots only	0.20
s - stems only	0.21
wsr - words, stems, and roots	0.23
rg2 – root bigram	0.05
wg3 – word trigram	0.24



Official monolingual runs (Ad hoc)	
Run	**Mean Avg. Precision**
aa - Automatic Arabic	0.22
ma - Manual Arabic	0.29
Post hoc runs (all automatic)	
w - words only	0.22
r - roots only	0.28
s - stems only	0.29
wsr - words, stems, and roots	0.28
rg2 – root bigram	0.15
wg3 – word trigram	0.31



1.5 Discussion

The results point to a few important conclusions:

1. The translation technique used was effective. In fact, for the official results the mean average precision of the non-expanded CLIR run was 89% relative to the mean average precision of the automatic Arabic run. Also, none of the CLIR runs were significantly better or worse than any of the Arabic run.
2. For the official runs, the results of individual queries were better than the median in 10 queries and 18 queries for the automatic non-expanded monolingual and cross language runs respectively.
3. Perhaps the use of n-grams for roots may have hurt the monolingual result.. When we tried using roots only as the index terms in later monolingual experiments, the resulting mean average precision was significantly better than any of our official results. However, the CLIR results were slightly hurt, but not significantly by the use of n-grams. Also using bigrams for roots seems to be a bad idea especially for CLIR runs.
4. The use of word character trigrams and stems produced the best results among the post-hoc experiments. Perhaps other experiments examining the effect of indexing using other n-grams, terms produced by morphological analysis, or combinations of both are warranted.

1.6 CLIR References

[1] Abu-Salem, Hani, Mahmoud Al-Omari, and Martha Evens, "Stemming Methodologies Over Individual Query Words for Arabic Information Retrieval." JASIS. 50 (6): 524-529, 1999.

[2] Al-Areeb Electronic Publishers, LLC. 16013 Malcolm Dr., Laurel, MD 20707, USA

[3] Al-Kharashi, Ibrahim and Martha Evens, "Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval." JASIS. 45 (8): 548-560, 1994.

[4] Beesley, Kenneth, "Arabic Finite-State Morphological Analysis and Generation." COLING-96, 1996.

[5] Beesley, Kenneth, Tim Buckwalter, and Stuart Newton, "Two-Level Finite-State Analysis of Arabic Morphology." Proceedings of the Seminar on Bilingual Computing in Arabic and English, Cambridge, England, 1989.

[6] Darwish, Kareem, "Building a Shallow Morphological Analyzer in One Day", www.glue.umd.edu/~kareem/hamlet/arabic/sebawai.tar.gz

[7] MacIntyre, Robert, "North American News Text Supplement", LDC98T30, LDC, 1998.

[8] NIST, Text Research Collection Volume 5, April 1997.

[9] tarjim.ajeeb.com, Sakhr Technologies, Cairo, Egypt www.sakhr.com

2 Video Track

2.1 Introduction

Our primary focus in this track was to get exposure to the process, test existing algorithms and determine the types of queries our current approaches was suited for. As previously stated, we participated in both the shot boundary detection and known item search.

2.2 Shot Boundary Detection

2.2.1 Overview

There has been a tremendous amount of work done on problem of "shot" detection in video. Our system was originally developed and extended in 1995 to process large quantities of MPEG-compressed video and provide a visual summary. In order to provide such a summary, we originally defined a shot change not only as a cut or gradual change, but also as the point where a significant amount of new content was introduced in the scene, either by new subjects appearing, or the camera panning to a new view of the current environment. The system runs at about 3x real-time and relies on a consistent and predictable coding of the video.

2.2.2 Approach

MERIT [21] detects cut shot changes by examining the MPEG macroblocks and DCT coefficients. If macroblocks of a frame rely very little on the proceeding or succeeding frames for encoding, the likelihood is high that there is shot change since same shot frames use the same information. Shot changes are determined by calculating the faction of macroblocks using information from other frames' macroblock to the total number of macroblocks. If this faction is below a threshold then is the potential for a shot change. All self-encoded frames are considered a potential for further processing the validation phase if it comes directly after a previous potential frame. In the validation phase DCT values of potential shot changes frames are decoded. If there is sufficient change in the DCT values, then the shot change is kept in the results. A shot change is validated by determining the difference between the DCT values of the frames before and after the potential frame. If the difference is above a threshold then it is considered a shot change. The thresholds for the system were determined by separately testing 12 minutes of video data of various genres (animations, commercials, movies, news, sports, and surveillance) that minimized the number of false and undetected transitions. No training was done with TREC collection. Details of the algorithm can be found in [21]. The MERIT system is available upon request to research organizations.

Gradual scene change detections are detected by projecting the DCT coefficient feature vector into a low dimensional space using a linear time reduction algorithm know as FastMap. The layout of these low dimensional points are tracked and if they do not cluster, a gradual change is detected. Details can be found in [22].

2.2.3 Experiments

Overall the system performed worse than the weighted median in all performance categories (Figure 1a). Incorrect cut transition had a severe impact on the overall results (Figure 1b). In gradual shot scene detection, the system performed better but missed more than the median performance system (Figure 1c). The system achieves its best results with database videos (ahf1, eal1, pfm1) with a bit-rate of 1.4MB/sec. Although some videos (ann. i005, anni009) had higher bit-rates, the grainy quality of the video degraded the accuracy of the macroblocks and DCT coefficients. Database clips with lower bit rates had lower performance rates.

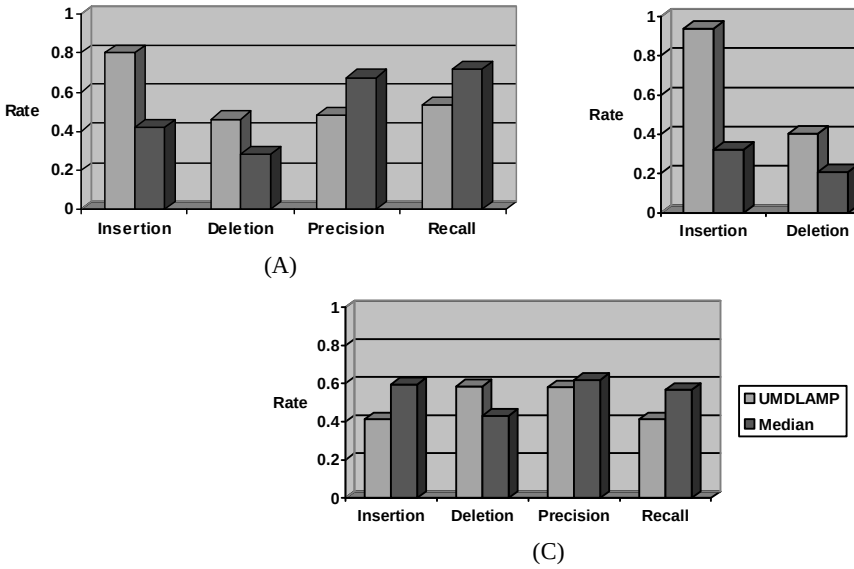


Figure 1: Overall Performance

2.2.4 Discussion

An examination of the undetected transitions indicates that reliance of DCT values for validation of shot changes makes it difficult to detect certain shot change situations. Cut transitions in which the two scenes were very similar in color or very dark were problematic. For example, in BOR08.mpg, there where many shot changes involving transition between old photographs that were prominently gold in color. Since the difference in the DCT would be minimal the difference did not produce a value above the threshold to indicate a shot change. This problem also occurred with gradual transitions that involved fades to black, fades from black, similar color or were dark in nature. The difference between frames did not produce a value greater than the threshold since the DCT values of the dark areas dominated varied very little from each other.

Our system often found transitions in clips where there were none.. The situations in which this error occurred were typically either camera jitter, when the camera made a sudden movement in a new direction, when a background object moved into a prominent position in the foreground, or when the camera zoomed in on a object. The macroblocks indicate a large change and was confirmed in the validation process since was a substantial change in color.

Although working with in the MPEG compressed domain is a quick way to analyze video it can produce errors. The reliance on DCT values makes it difficult to detect transitions that involve scenes that are dark or have similar prominent color scheme. In these cases the probability is high that the changes will not register above a threshold. In the future an adaptive threshold is needed to detect the presence of these situations during the validation phase.

2.3 Known Item and General Search

2.3.1 Overview

Content-based retrieval has been subject to active research since the early 1990's [3] and a large number of experimental image retrieval systems have been introduced, such as BlobWorld [4], Chabot [5], Mars [6], NeTra [7], Photobook [8], QBIC [9], Surfimage [10] and VisualSEEK [11]. These systems retrieve images based on cues such as color, texture, and shape, of which color remains as the most powerful and most useful feature for general purpose retrieval. Color-based retrieval first evolved from simple statistical measures such

as average color to color histograms [9,5,8], but histograms alone suffer for large collections since different configurations can produce the same histogram.

2.3.2 Approach

The spatial correlation of colors as a function of spatial distance is an image feature introduced by Huang *et al.* [12] known as a correlogram. Our approach extends this method and uses a novel color content method, the Temporal Color Correlogram (TCC), to capture the spatio-temporal relationship of colors in a video shot using co-occurrence statistics. TCC is an extension of HSV Color Correlogram (CC), which is found very effective in content-based image retrieval [1]. Temporal Color Correlogram computes autocorrelation of the quantized HSV color values from a set of frame samples taken from a video shot. In this paper, the efficiencies of TCC and HSV Color Correlogram (CC) are evaluated against other retrieval systems participating VideoTREC track evaluation. Tests are executed using our retrieval system, CMRS, which is specifically developed for multimedia information retrieval purposes.

2.3.2.1 Correlogram extension in temporal domain

In digital video, color and intensity information change temporally over a shot, creating the illusion of object or observer movement. This knowledge is also used in modern video compression algorithms, where motion is estimated by moving rectangular blocks of quantized illumination and colors towards the expected direction of motion (MPEG) [19]. In order to create a content-based descriptor for a video shot, such structural information should be trasferred into computable features.

The temporal changes of video shot contents can be described using the temporal correlogram. The benefits over more traditional approaches, such as color histograms, derive from its ability to encapsulate the temporal changes in small spatial color environments. Figure 2 depicts a temporal color change in a small spatial environment. Whereas the color histogram would only portray the proportional amount of color in these frames, temporal correlogram will capture information about the spatial changes of these colors occurring over time.

Figure 2: Temporal color change illustrated by frame sequence. Temporal correlogram captures the dispersion of the color element whereas histogram does not.

Let N be the amount of sample frames I^n taken from a shot S. Values of n vary from 1 to N indicating the index in the sample frame sequence. The temporal correlogram is calculated as

$$\bar{\gamma}_{c_i,c_j}^{(d)}(S) \equiv \Pr_{p_1 \in I_{c_i}^n,\, p_2 \in I^n}\left[p_2 \in I_{c_j}^n \,\middle|\, |p_1 - p_2| = d\right] \qquad (3)$$

which gives the probability that given any pixel p_1 of color c_i, a pixel p_2 at a distance d from the given pixel p_1 is of color c_j among the shot's sample frames I^n.

For computational benefits [1], the Temporal Color Correlogram (noted here as TCC) used for this study is computed as an autocorrelogram, which is obtained from Eq. 3. by replacing c_j with c_i. The quantization of HSV color space for TCC follows the quantization of CC.

2.3.3 Experiments

To evaluate the temporal correlogram efficiency, we used 11 hours database of MPEG1 videos available for VideoTREC track participants [2]. First, the video material was segmented to create shots using VideoLogger video editing software from Virage [20] and our own system (above) but the Virage results were used. For the 11 hours of video, 7375 shot segments were created with the average shot length of approximately 5 seconds. From the shot frames, the beginning frame was selected as a representative key frame, from which the static image feature, CC, was obtained. In order to calculate TCC non-exhaustively and to keep the number of samples in equal for varying shot lengths, each shot was sampled evenly with a respective sampling delay so that the number of sample frames did not exceed 40. After segmentation, shot features were fed into our CMRS retrieval system and queries were defined using either example videos or example images depending on the respective VideoTREC topic specification [2].

VideoTREC result submission contained retrieval results of two system configurations. First configuration was obtained using TCC for the retrieval topics that contained video examples in the topic definition. Second configuration used CC for topics that contained example images in their definition. Table 6 shows the average precisions of General Search results for the TCC feature in different topic categories. As the results show, TCC as a purely automatic method did worse in Interactive and Automatic+Interactive topics, since no other cues than this color structure feature were used in a query (meaning there was no human involvement to prune the results). The General Search overall results were not impressive in contrast to other participating systems as can be seen from the Figure 3 that depicts all system precisions ranked into evolving curve starting from worst system on the left. However, the average precision in Automatic topics ranks TCC higher, just below the median of all systems.

Topic Type (# of topics)	Average Precision
Interactive (3)	0.08
Automatic+Interactive (8)	0.13
Automatic (17)	0.24
Overall Average (28)	0.19

Table 6:.General Search Results for TCC with different topic categories.

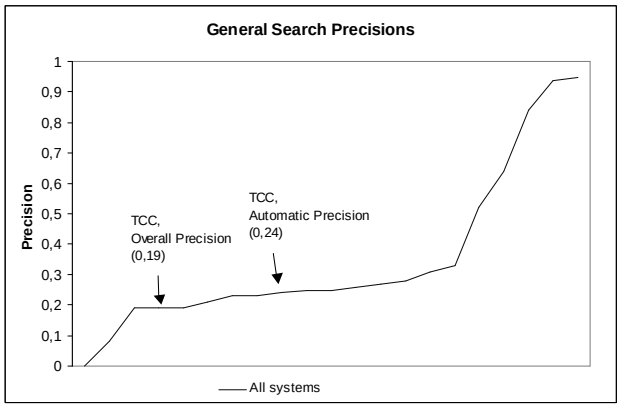


Figure 3: TCC General Search performance against other systems. The curve indicates ranked list of system precisions, worst being in the left and best in the right end of the curve.

Table 7 shows the Known Item Search results with different match parameters. The parameters define when a retrieved item is a successful match to a known item. The loosest criteria (0.333/0.333) for the match expects the retrieved shot to be overlapping with known item at least one third of the shot duration having the same rule for the known item sequence. The tightest criteria require two thirds of the shot durations to overlap. It can be seen that the results for the CC configuration are dismal whereas TCC is doing better. In Figure 4 one can see that the TCC recall is ranked in the median (11th out of 21) of all systems for Known Item Searches.

RECALL

Parameters	**0.333/0.333**	**0.333/0.666**	**0.666/0.333**	**0.666/0.666**
TCC	0,181	0,117	0,07	0,02
CC	0,014	0,002	0,014	0,001

PRECISION

Parameters	**0.333/0.333**	**0.333/0.666**	**0.666/0.333**	**0.666/0.666**
TCC	0,011	0,005	0,005	0,001

Table 7: Recall and Precision averages for TCC and CC configurations with different match parameters.

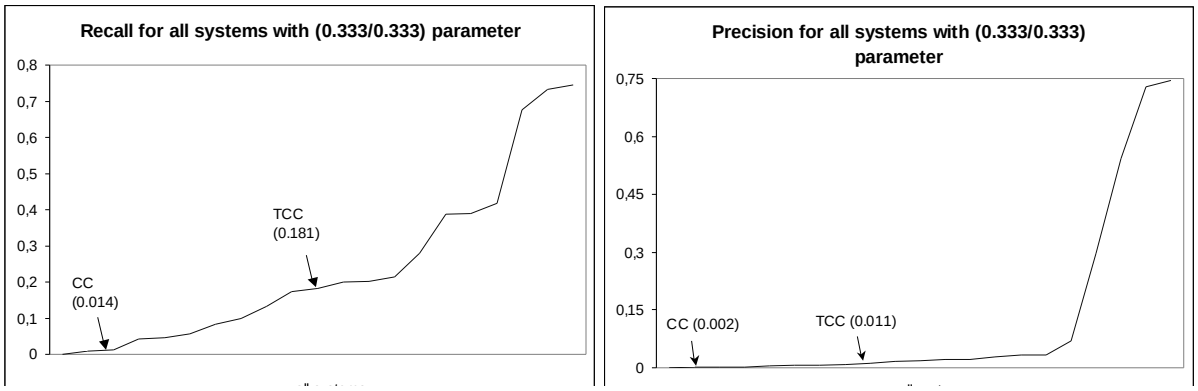


Figure 4: TCC and CC recall and precision against respective values of other systems. The curve indicates ranked list of system precisions/recalls, worst being in the left and best in the right end of the curve.

Table 8 shows the top 5 topics for the Known Item Searches. Topic 3 was the most successful. It considered finding video segments that depict a lunar vehicle traveling on the moon. Other topics in the list considered problems of finding a yellow boat, snow capped mountains or a student from classroom footage.

	Precision	Recall
1st	Topic 3	Topic 3
2nd	Topic 35	Topic 6
3rd	Topic 36	Topic 35
4th	Topic 4	Topic 36
5th	Topic 6	Topic 4

Table 8: Top 5 topic results by precision and recall

2.3.4 Discussion

The semantic gap is too large for video analysis features like TCC and CC in search problems such as in the General Search topics and the topics containing an example image. In other words, the structural, 'mechanical', content of example images and video shots doesn't convey the meaning of the semantic request that the person defining a query actually pursues. This can be improved by combining other cues such as audio and text to focus the search towards more meaningful locations in a video.

Better results were obtained in the topics that seek Known Items with similar structural shot properties. Topics that tried to locate footage from the same target with different camera angles or object positions gave the most successful results. The evaluation criteria for a search hit was rather strict. It leaves many questions whether people searching video databases want the exact locations of the known items returned, or rather, just a pointer inside a video where one can start to examine the video by himself. Is it more beneficial to provide the user with accurate segments together with very low retrieval ranks, rather than giving less accurate results with higher ranks? Surely users will rather watch a couple of longer segments from the top ranks than to wade through tens of useless clips in order to find the exact match with low rank. What makes the problem worse is that no automatic system will be accurate enough to successfully encapsulate the varieties in semantic definitions that people will use in their queries into heterogeneous video databases. In the retrieval results there will always exist loads of useless segments among the really significant ones.

2.4 Video References

[1] Ojala T, Rautiainen M, Matinmikko E & Aittola M (2001) Semantic image retrieval with HSV correlograms. Proc. 12th Scandinavian Conference on Image Analysis, Bergen, Norway, 621-627.

[2] TREC-2001 Video Retrieval Track Home Page, (10/25/2001) http://www-nlpir.nist.gov/projects/t01v/t01v.html

[3] Eakins J & Graham M (1999) Content-Based Image Retrieval: A report to the JISC Technology Applications Programme. Institute for Image Data Research, University of Northumbria at Newcastle, United Kingdom. *http://www.unn.ac.uk/iidr/research/cbir/report.html*.

[4] Carson C, Belongie S, Greenspan H & Malik J (1997) Region-based image querying. Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries, San Juan, Puerto Rico, 42-49.

[5] Ogle V & Stonebraker M (1995) Chabot: retrieval from a relational database of images. IEEE Computer Magazine 28:40-48.

[6] Ortega M, Rui Y, Chakrabarti K, Porkaew K, Mehrotra S, Huang TS (1998) Supporting ranked Boolean similarity queries in MARS. IEEE Transactions on Knowledge and Data Engineering 10:905-925.

[7] Ma WY & Manjunath BS (1997) NeTra: a toolbox for navigating large image databases. Proc. International Conference on Image Processing, Santa Barbara, CA, 1:568-571.

[8] Pentland A, Picard R & Sclaroff S (1996) Photobook: content-based manipulation of image databases. International Journal of Computer Vision 18:233-254.

[9] Flickner M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, Gorkani M, Hafner J, Lee D, Petkovic D, Steele D & Yanker P (1995) Query by image and video content: The QBIC system. IEEE Computer Magazine 28:23-32.

[10] Mitschke M, Meilhac C & Boujemaa N (1998) Surfimage: a flexible content-based image retrieval system. Proc. Sixth ACM International Conference on Multimedia, Bristol, UK, 339-344.

[11] Smith J & Chang S-F (1996) VisualSEEK: a fully automated content-based image query system. Proc. Fourth ACM International Conference on Multimedia, Boston, MA, 87-98.

[12] Huang J, Kumar SR, Mitra M & Zhu WJ (1997) Image indexing using color correlograms. Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 762-768.

[13] Pass G, Zabih R & Miller J (1996) Comparing images using color coherence vectors. Proc. Fourth ACM International Conference on Multimedia, Boston, MA, 65-73.

[14] Huang J, Kumar SR, Mitra M & Zhu WJ (1998) Spatial color indexing and applications. Proc. Sixth International conference on Computer Vision, Bombay, India, 602-607.

[15] Huang J, Kumar SR & Mitra M (1997) Combining supervised learning with color correlograms for content-based image retrieval. Proc. Fifth ACM International Conference on Multimedia, Seattle, WA, 325-334.

[16] Ma WY & Zhang HJ (1998) Benchmarking of image features for content-based retrieval. Proc. 32nd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 1:253-257.

[17] Hafner J, Sawhney HS, Equitz W, Flickner M, Niblack W (1995) Efficient color histogram indexing for quadratic form distance functions. IEEE Transactions on Pattern Analysis and Machine Intelligence 17:729-736.

[18] Tekalp AM (1995) Digital video processing. Prentice Hall signal processing series, US, 526.

[19] MPEG.ORG (10/25/2001) http://www.mpeg.org/MPEG/video.html

[20] Virage, Inc. (10/25/2001) http://www.virage.com/

[21] V. Kobla, D.S. Doermann, and A. Rosenfeld, Compressed domain video segmentation, CfAR Technical Report CS-TR-3688, University of Maryland, College Park, 1996.

[22] V. Kobla, D. DeMenthon, and D. Doermann, Special effect edit detection using VideoTrails: a comparison with existing techniques, Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases VII, Jan, 1999.