Lazy Users and Automatic Video Retrieval Tools in (the) Lowlands

The Lowlands Team
CWI[1], TNO[2], University of Amsterdam[3], University of Twente[4]
The Netherlands

This work was funded (in part) by the ICES/KIS MIA project and the Dutch Telematics Institute project DRUID. The following people have contributed to these results (appearing in alphabetical order): Jan Baan[2], Alex van Ballegooij[1], Jan Mark Geusenbroek[3], Jurgen den Hartog[2], Djoerd Hiemstra[4], Johan List[1], Thijs Westerveld[4], Ioannis Patras[3], Stephan Raaijmakers[2], Cees Snoek[3], Leon Todoran[3], Jeroen Vendrig[3], Arjen P. de Vries[1] and Marcel Worring[3].

1 Introduction

This paper describes our participation in the TREC Video Retrieval evaluation. Our approach uses two complementary automatic approaches (the first based on visual content, the other on transcripts), to be refined in an interactive setting. The experiments focused on revealing relationships between (1) different modalities, (2) the amount of human processing, and (3) the quality of the results.

We submitted five runs, summarized in Table 1. Run 1 is based on the query text and the visual content of the video. The query text is analyzed to choose the best detectors, e.g. for faces, names, specific camera techniques, dialogs, or natural scenes. Query by example based on detector specific features (e.g. number of faces, invariant color histograms) yields the final ranking result.

To assess the additional value of speech content, we experimented with a transcript generated using speech recognition (made available by CMU). We queried the transcribed collection with the topic text combined with the transcripts of video examples. Despite of the error-prone recognition process, the transcripts often provide useful information about the video scenes. Run 2 combines the ranked output of the speech transcripts with (visual-only) run 1 in an attempt to improve its results; run 3 is the obligatory transcript-only run.

Run	Description
1	Detector-based, automatic
2	Combined 1–3, automatic
3	Transcript-based, automatic
4	Query articulation, interactive
5	Combined 1–4, interactive, by a lazy user

Table 1: Summary of runs

Run 4 models a user working with the output of an automatic visual run, choosing the best answer-set from a number of options, or attempting to improve its quality by helping the system; for example, finding moon-landers by entering knowledge that the sky on the moon is black or locating the Starwars scene by pointing out that the robot has golden skin.

Finally, run 5 combines all information available in our system: from detectors, to speech transcript, to the human-in-the-loop. Depending on the evaluation measures used, this leads to slightly better or slightly worse results than using these methods in isolation, caused by laziness expressed in the model for selecting the combination strategy.

2 Detector-based Processing

The main research question addressed in run 1 was how to make query processing fully automatic. This includes devising mechanisms that bridge in an automatic way the semantic gap [13] between (1) the user's information need as specified on the one hand by the topic text description and on the other hand by the video and image examples and (2) the low level features that can be extracted from the video. We propose a unifying approach in which a wide range of detectors and features are combined in a way that is specified by semantic analysis of the topic description. Section 2.1 describes the system's architecture and Section 2.2 the specific detectors and features used.

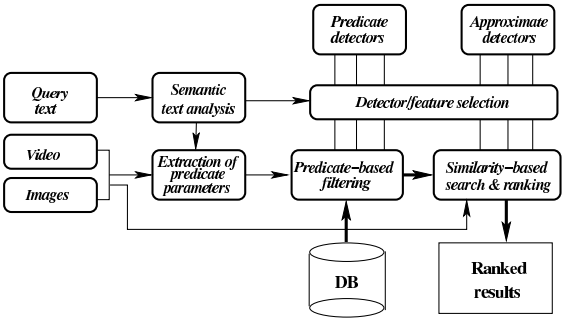

Figure 1: Architecture for automatic system.

2.1 System's architecture

A great challenge in automatic retrieval of multimedia material is to determine which aspect of the information carried in the audiovisual stream is relevant for the topic in question. The aspects of information that we restrict to are determined by the specific detectors that our systems employs. Examples are color-based detectors, face detectors or modules that detect the camera technique or the presence of monologues.

In order to select the relevant detectors we associate them with concepts that exist in the 'is-a' hierarchy of the Wordnet dictionary. For example, the face detectors are associated with the concept *'person, individual, human'*. In order to determine if the specific detector is to be used for a topic, we analyze its text[1] in two steps. In the first step, a syntactic analysis discards the words that are not nouns, verbs or adjectives. In the second step, we feed the remaining words to the Wordnet dictionary and detect if the concepts that are associated with the detectors that we have at our disposal are present in the 'is-a' hierarchy of the most common meaning of the words in question. Such an approach makes good associations for most of our detectors. However, it exhibits its limitations in the case that the query word has also other meanings. For example the most common meaning of the word "pan" is *cooking utensil, cookware"*. Such ambiguities are resolved in our current system by maintaining an additional set of keywords for the camera motion detector.

Once the appropriate set of detectors are selected we proceed to the retrieval of the relevant video clips. In order to do so we need to make a distinction between two different kinds of detectors[13]:

- detectors for *exact* queries that yield a yes/no answer depending if a set of predicates is satisfied (e.g. does the camera exhibit a zoom-in?). The face detector, the monologue detector, and the camera technique detector fall in this category;
- detectors for *approximate* queries that yield a measure that expresses how similar is the examined video clip with an example video clip. In this category fall the module for color-based retrieval.

The selected detectors of the first category are used to filter-out irrelevant material. Then, a query-by-example based search on the (selected) detectors of the second category produces the final ranked results. In case that the analysis of the topic description determines that no detector of the second category should be selected, the ranking is based on the shot length.

Let us finally note that some of the detectors of the first category learn some of their parameters from the examples provided in the topic. Such a detector is the face detector which learns from the query example how many persons should appear in a video clip so that it is characterized as relevant.

2.2 Detectors

Another goal in the evaluation was to assess the quality of the detectors discussed in this Section. The results of run 1, in the cases that the right detector was chosen, indicate the techniques perform with fairly high precision.

2.2.1 Camera technique detection

To detect the camera technique used in a shot, we use a method based on spatiotemporal slices of the original video to detect whether the apparent motion is due to known camera activities such as pan and tilt, or the scene is static [9]. In the former case, we estimate the percentage of the apparent motion that is due to camera's pan, tilt and zoom (e.g. 60% zoom, 5% tilt and 35% pan). Clips to which the dominant apparent motion is not caused by camera operations are characterized as "unknown".

The detector of the camera technique was used for topics 44, 48 and 74 in which the keywords 'zoom' and 'pan' appear. The system categorized successfully apparent motions that are due to pure camera operations (90% precision for topic 44 and 100% precision for query 74), but failed for topic 48 in which the zooming-in is not due to change in camera's focal-length. The reason for the latter is that the apparent motion field depends on the distance between camera and scene.

[1] We analyzed only the first sentence of the topic description.

2.2.2 Face detector

An off-the-shelf face detector (Rowley[12]) is used in order to detect how many faces are present in the video clip in question. The result is compared with the number of faces that were detected in the image example. We use five categories of numbers of faces: 'no-face", '1-face', '2-faces', '3-faces', 'many-faces'. The face detector is associated with the general concepts *"person, individual, human"* and *"people"* for the Wordnet hierarchy. It works well for topics requesting humans appearing in (near) frontal view (e.g. 100% precision for topic 41) but, naturally, is not relevant otherwise (e.g. water-skier in topic 31).

2.2.3 Caption retrieval

For finding given names in the visual content, three steps are taken:

- text segmentation;
- OCR;
- fuzzy string matching.

For text segmentation of video frames we use a dual approach. The first approach is a color segmentation method [20], to reduce the number of colors, while preserving the characters. The second approach is intensity based, using the fact captions are superimposed. OCR is done by ScanSoft's TextBridge SDK 4.5 library [16]. Finally, string matching is done using k-differences approximate string matching (see e.g. [1]).

The detector worked well in retrieving video based on the text that appears as caption. It has been applied for 24 topics that contain capitalized text (e.g. 'House' and 'Congress' in topic 30) with around 10% and 20% false positives and false negatives respectively. However, the retrieved video (even if it contained the query text as a caption) did not always match with the user's intention (e.g. the result for topic 30 is a shot of a text document). Therefore, we have used the results of such a detector only when the topic consists of a text description only (i.e. no media example is available). Only in that case the shots that are retrieved based on this detector are used to initiate a color-based query.

2.2.4 Monologue detection

The method for monologue detection [15] first uses a camera distance heuristic based on Rowley's face detector [12]. Only shots showing faces appearing in front of the camera within a certain distance are processed. In a post-processing stage all those shots are checked upon using three constraints:

- shot should contain speech;
- shot should have a static or unknown camera technique;
- shot should have a minimum length.

When all constraints are met, a shot is classified as a monologue. Subsequently, the selected shots are ranked based on their length: the longer the shot the higher the likelihood of it being a true monologue.

This detector has been used for topics 40, 63 and 64 with a very good performance (near 100% precision). The performance is lower for topic 64 (60% precision), because satisfying the information need (*male* interviewees) requires to distinguish between sexes, a predicate not anticipated in our current system.

2.2.5 Detectors based on color invariant features

Ranking of the shots remaining after filtering using predicate detectors, was accomplished by implementing a query by image example paradigm. For each keyframe a robust estimate of the color content of each keyframe is computed by converting the keyframe to the Gaussian color model as described in [4]. The Gaussian color model is robust against spatial compression noise, achieved by the Gaussian smoothing involved. Further, the Gaussian color model is an opponent color representation, for which the channels are largely uncorrelated. Hence, the color histograms can be constructed as three separate one-dimensional histograms. The keyframes were stored in a database, together with their color histogram information. Matching of example keyframe against the database targets is efficiently performed by histogram intersection between each of the three (one-dimensional) histograms. Matching time was within a second, ensuring system response to be adequate for interactive retrieval purposes.

3 Probabilistic Multimedia Retrieval

This section introduces our probabilistic approach to information retrieval, an approach that unifies models of discrete signals (i.e. text) and models of continuous signals (i.e. images) into one common framework. We usually take for text retrieval an approach based on statistical language models [6, 7, 10, 3], which uses a mixture of discrete probability measures. For image retrieval, we experimented with a probabilistic model that uses a mixture of continuous probability measures [18].

The basic model can in principal be used for any type of documents and queries, but for now we assume our documents are shots from a video. In a probabilistic setting, ranking the shots in decreasing order of relevance amounts to ranking the shots by the probability $P(Shot_i|Q)$ given that query. Using Bayes' rule we can rewrite this to:

$$\begin{aligned} P(Shot_i|Q) &= \frac{P(Q|Shot_i)P(Shot_i)}{P(Q)} \\ &\propto P(Q|Shot_i)P(Shot_i) \end{aligned}$$

In the above, the right-hand side will produce the same ranking as the left-hand side. In absence of a query, we assume that each shot is equally likely of being retrieved, i.e. $P(Shot_i) = \text{constant}$. Therefore, in a probabilistic model for video retrieval shots are ranked by their probability of having generated the query. If a query consists of several independent parts (e.g. a textual Qt and visual part Qv), then the probability function can be easily expressed as the joint probability of the different parts. Assuming independence between the textual part and the visual part of the query leads to:

$$P(Q|Shot_i) = P(Qt|Shot_i)P(Qv|Shot_i) \quad (1)$$

3.1 Text retrieval: the use of speech transcripts

For text retrieval, our main concern was adapting our standard language model system to the retrieval of shots. More specifically, we were interested in an approach to information retrieval that explicitly models the familiar hierarchical data model of video, in which a video is subdivided in scenes, which are subdivided in shots, which are in turn subdivided in frames.

Statistical language models are particularly well-suited for modeling complex representations of the data [6]. We propose to rank shots by a probability function that is a linear combination of a simple probability measure of the shot, of its corresponding scene, and of the corresponding video (we ignore frames, because in practice words in transcribed speech are not associated with a particular frame).

Assuming independence between query terms:

$$\begin{aligned} P(Qt_1, \quad \cdots \quad , Qt_n|Shot) &= \\ \prod_{j=1}^{n} \quad & (\pi_1 P(Qt_j) + \pi_2 P(Qt_j|Video) + \\ & \pi_3 P(Qt_j|Scene) + \pi_4 P(Qt_j|Shot)\,) \end{aligned}$$

In the formula, $Qt_1, \cdots, Qt_n$ is a textual query of length n, $\pi_1, \cdots, \pi_4$ are the probabilities of each representation, and e.g. $P(Qt_j|Shot)$ is the probability of occurrence of the term Qt_j in the shot: if the shot contains 10 terms in total and the query term in question occurs 2 times then this probability would be simply $2/10 = 0.2$. $P(Qt_j)$ is the probability of occurrence of the term Qt_j in the collection.

The main idea behind this approach is that a good shot is one that contains the query terms; one that is part of a scene that has more occurrences of the query terms; and one that is part of a video that has even more occurrences of the query terms. Also, by including scenes in the ranking function, we hope to retrieve the shot of interest, even if the video's speech describes the shot just before it begins or just after it finishes. Depending on the information need of the user, we might use a similar strategy to rank scenes or complete videos instead of shots, that is, the best scene might be a scene that contains a shot in which the query terms (co-)occur.

3.2 Image retrieval: retrieving the key frames of shots

For the visual part, we cut the key frames of each shot into blocks of 8 by 8 pixels. On these blocks we perform the Discrete Cosine Transform (DCT), which is used in the JPEG compression standard. We use the first 10 DCT-coefficients from each color channel[2] to describe the block. If an image consists of n blocks, we have n feature vectors describing the image (each vector consisting of 30 DCT coefficients). Now the probability that a particular feature vector (Qv_j) from our query is drawn from a particular shot ($Shot_i$) can be described by a Gaussian Mixture Model [18]. Each shot in the collection is then described by a mixture of C Gaussians.[3] The probability that the a query (Qv) was drawn from $Shot_i$ is simply the joint probability for all feature vectors from Qv. We assume independence between the feature vectors

$$\begin{aligned} P(Qv_1, \quad \cdots \quad , Qv_n|Shot_i) = \\ \prod_{j=1}^{n} \quad \sum_{c=1}^{C} \pi_{i,c}\mathcal{G}(Qv_j, \mu_{i,c}, \Sigma_{i,c}) \end{aligned} \quad (2)$$

where $\pi_{i,c}$ is the probability of class c from $Shot_i$ and $\mathcal{G}(Qv_j, \mu_{i,c}, \Sigma_{i,c})$ is the Gaussian density (or normal density) for class c from shot i with mean vector μ_i and co-variance matrix Σ_i. If m is the number of

[2] We work in the YCbCr color space.
[3] We used a mixture of 8 Gaussians.

DCT features representing a shot, the Gaussian is defined as:

$$\mathcal{G}(x, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^m |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (3)$$

For each of the shots in the collection we estimated the probability, mean and co-variance for each of the Gaussians in the model using the Expectation Maximization algorithm [11] on the feature vectors from the shots.

At this stage, equation 2 could be used to rank shots given a query, however, its computational complexity is rather high. Therefore, instead of this Feature Likelihood (the likelihood of drawing all query features from a shot model) we computed the Random Sample Likelihood introduced by Vasconcelos [18]. The Random Sample Likelihood is defined as the likelihood that a random sample from the query model was drawn from the shot model, which comes down to building a model for your query image(s) and comparing that model to the documents models to rank our shots.

3.3 Experimental setup

For the textual descriptions of the video shots, we used speech transcripts kindly provided by Carnegie Mellon University. Words that occurred within a transition between two shots were put within the previous shot. We did not have a division of the video into scenes, nor did we build a scene detector. Instead, scenes were simply defined as overlapping windows of three consecutive shots. Because we did not have material available to tune the model, the values of the parameters were determined on a ad-hoc basis. Instead of implementing the model as described, we took a more straightforward approach of doubling artificially the terms in the middle shots to obtain pseudo-documents, and ranked those using the 'standard' model with parameter $\lambda = 0.15$ (see [6]). For the queries, we took both the words from the textual description of the topics and the words occurring in the video examples' time frame, if these were provided.

Run 2 combines automatically the results of run 1 and run 3. It is produced by applying the ranking strategy determined by query analysis to the results of the speech transcript run, using the latter as a filter; unless query analysis decides the transcripts would be irrelevant. Transcripts are ignored if the video is not expected to contain query words, which is the case of predicate detectors like camera motion techniques and monologues.

Run	R@100	P@100
Text-based (run 3)	0.133	0.007
Detector-based (run 1)	0.101	0.003
Image-based (unofficial)	0.065	0.003
Combined (run 2)	0.085	0.005
Combined (unofficial)	0.079	0.005

Table 2: Recall @ 100 and precision @ 100 for probabilistic runs

The results of run 2 did not improve upon run 3, which may be attributed to the ad-hoc approach of combining methods. This motivated additional experiments with a pure probabilistic approach. We evaluated this alternative on the known item search task in an unofficial run. Table 2 compares these unofficial results with our submitted runs. A returned fragment is regarded relevant if the intersection between the fragment and a known item contains at least one third of the fragment and one third of the known item.

Unfortunately, the unofficial combined run is not better than run 2. The difference between measured performance of the unofficial image-based run and run 1 may have influenced this result. Although it is too early to draw strong conclusions from our experiments, another plausible explanation is that the assumption of independence between the textual and visual part is not a valid one.

4 Interactive Experiments

Our interactive topic set consisted – by mistake – of only 30 topics, of which we 'solved' 9, and could not produce any answer for 2.[4] This Section presents mostly positive highlights of our work on the interactive topics for the Video Collection. Note that our interactive users do not identify the correct answers in the retrieved result sets, so precision is not expected to be 100% (see also Section 5).

A quick investigation of behavior of 'standard' image and video analysis techniques on the interactive topics proved our suspicion that purely automatic systems cannot be expected to perform well on most topics: a result of the 'difficult' queries (not just 'sunset' and 'tropical fish') and the low quality of the video data itself. Thus, we focused on the research question how users could improve upon naive

[4]The slightly smaller topic set used was the result of missing a crucial message on the mailing list.

Figure 2: Topic 33, White fort, example(left) and known-item(right) keyframes.

query-by-example methods to express their information needs in a more successful manner.

The retrieval system used for this task is developed on top of Monet, a main-memory database system. It uses a variety of features that are all based on the distribution of color in the keyframes of the shots. Details on the particular features used are provided in a forth-coming technical report [17]. Note that, even though we participated in the interactive topics, the lack of a proper user interface in our current implementation implies that system interaction consisted mostly of writing scripts in Monet's query language.

4.1 Color-based Retrieval Techniques

The results of topics 33 (White fort) and 54 (Glenn Canyon dam) clearly demonstrate that popular color-based retrieval techniques can indeed be successful, *as long as the query example is derived from the same source as the target objects*. Figure 2 shows the keyframes representing the example and known item for topic 33; any color-based technique worked out well for this query. Topic 54 was solved using a spatial color histogram retrieval method, implicitly enforcing locality such as blue sky on top, brown rocks on the sides and white water and concrete dam in the center.[5]

Topic 53 (Perseus) is an example where we were lucky: the example image provided happens to look surprisingly much like the Perseus footage in the data-set, and spatial color histogram retrieval retrieves a large number of Perseus clips.

Topic 24 (R. Lynn Bondurant) provides an interesting lesson about the balance between recall and precision using content-based retrieval techniques. Although it is relatively easy to find some other shots showing Dr. Bondurant – those where he sits in the same room wearing the same suit – finding *all* shots is a completely different question.

The other topics confirm our intuition that we should not expect too much from 'traditional' content-based retrieval techniques. Although more advanced features based on texture and shape possibly could help in solving more topics directly, we doubt whether a *significant* improvement over these results would be achieved. If available however, *domain-specific* detectors (such as the face detectors deployed in run 1) can provide good performance for specific tasks.

[5]Obviously, nothing guaranteed the dams found are indeed Glenn Canyon dams...

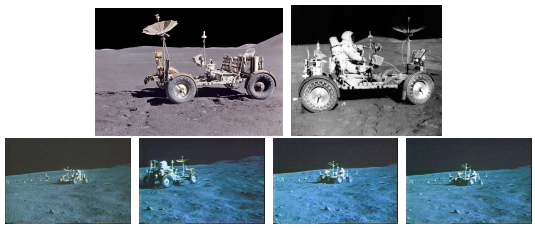

Figure 3: Topic 19, Lunar rover, examples (images on top) and the keyframes of the correct answers.

4.2 Query Articulation

As an alternative approach, we propose to put more emphasis on the quality of the queries expressing the underlying information need. We aim for the interactive refinement from initial, broad multi-modal examples into relatively precise search requests, in a process we have termed *query articulation* [2]. In essence, articulating a query corresponds to constructing a query-specific detector on-the-fly.

The idea of query articulation is best demonstrated through the idea of a 'color-set'. Users define color-sets interactively by selecting regions from the example images, possibly extending the implied color-set by adding similar colors. Unlike the binary sets introduced in VisualSEEK [14], we essentially re-quantize the color space in a smaller number of colors, by collapsing the individual elements of a color-set onto a single new color.

Topic 19: Lunar Rover

Topic 19 (Lunar Rover) provides 2 example images showing the lunar rover. The visual differences between the (grayish) sample images and (bluish) known-items (shown in Figure 3) explain why color-based retrieval techniques are not successful on this topic. Query articulation allows users to circumvent this problem, by making explicit their own world knowledge: in scenes on the moon, the sky is black. This can be expressed in terms of the system using two simple filters based on color-sets:

- 'Black Sky': The filter is realized by selecting those keyframes for which the top 25% of the

Figure 4: The 'dark' color-set as defined for topic 19, Lunar rover.

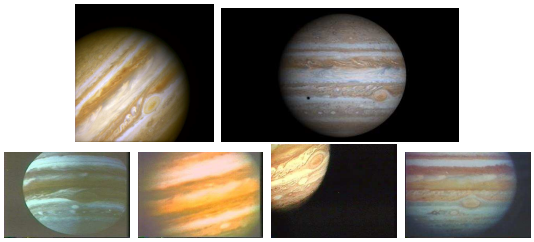

Figure 5: Topic 8, Jupiter, example (on top) and some correct answers keyframes.

image is at least 95% dark (a color-set shown in Figure 4).

- 'Non-black Bottom': making sure that no completely dark images are retrieved, (a large number of outer-space shots are present in the dataset) this second filter selects only those keyframes that do not have a black bottom as there should be lunar surface with the lunar rover visible. The filter is realized by selecting those keyframes for which the lower half of the image is less than 80% dark.

Together, these filters effectively reduce the total data-set of approximately 7000 keyframes to only 26, containing three of the four known items. Recall is improved using a follow-up query, ranking the images with a 'Black Sky' using the spatial color histogram method on a seed image drawn from the previous phase. This second step returns the four known items in the top-10.

Topic 8: Jupiter

The Jupiter topic is another example that benefits significantly from query articulation. At a first thought, this query may seem to be easy to solve, as planets have a typical appearance (a colored circle surrounded by black) and Jupiter should be easily recognized. But, examining the example images shown in Figure 5, it is apparent that colors in different photos of Jupiter can differ significantly.

An important characteristic of Jupiter is the distinguishable orange and white lines crossing its surface. Articulating this through color content, we decided to put emphasis on the orange content, the white content, and their interrelationships, expressed as filters on color-set *correlograms* [8]. Computing correlograms from the color-sets shown in Figure 6 produces 9-dimensional feature vectors, one dimension for each possible transition. To ensure that the results are not dominated by the auto-correlation coefficients, the resulting vectors are weighted using the inverse of their corresponding coefficients in the query images. The derived query finally finds some of the known-items, but recall remains low.

Another way to emphasize the striped appearance of Jupiter is to detect the actual presence of (horizontal) lines in images and rank the keyframes based on that presence. This was implemented by means of DCT-coefficients, classifying each DCT-matrix in the luminance channel of a keyframe into texture-classes. We used the classes 'horizontal-line', 'vertical-line', 'blank' and 'other'. The cheap method of ranking by simple statistics on these texture-classes proved only slightly worse than the previous (elaborate and expensive) method based on correlograms.

Although a combination of both results did not retrieve any additional answers, a minor improvement is obtained through a subsequent search, seeded with a retrieved shot found before.

Topic 25: Starwars

Finding the Starwars scene became a matter of honor, since we submitted the topic ourselves – perhaps a bit over-enthusiastically. After several unfruitful attempts using color histograms and color-sets, we decided to articulate the query by modeling the golden appearance of one of the robots, C3PO. This idea might work well, as we do not expect to find *many* golden objects in the data-set.

The appearance of gold does not simply correspond to the occurrence of a range of colors; its most distinguishing characteristic derives from the fact it is a *shiny* material, implying the presence of small, sharp highlights. We implemented two stages of boolean filters to capture these properties, followed by a custom ranking procedure.

The first filter selects only those images that

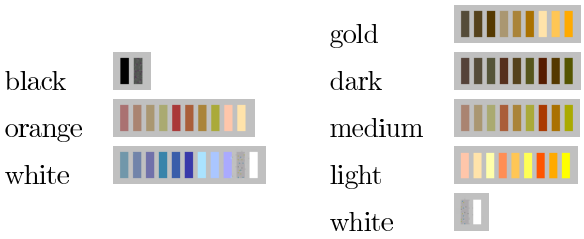

Figure 6: Color-sets used in the Jupiter (left) and the Starwars (right) topics.

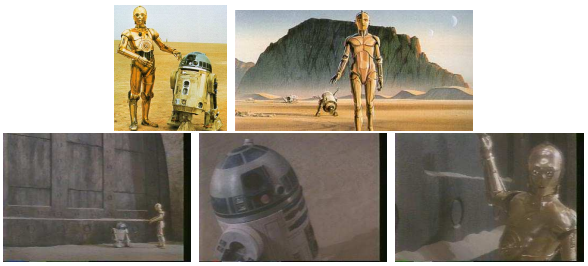

Figure 7: Topic 25, Starwars, examples(left 2 images) and the correct answers keyframes.

have sufficient amount of golden content. It checks whether images have at least 20% 'golden' pixels, using the gold color-set defined in Figure 6. Secondly, a set of filters reduces the data-set by selecting those images that contain the color(set)s shown, representing the appearance of gold in different lighting conditions, in a way expected for shiny metallic surfaces: a bit of white, some light-gold, a lot of medium-gold, and some dark-gold. Although the precise percentages to be selected are difficult to choose correctly, we believe the underlying idea is valid, as we modeled expected levels of gold-content for a shiny-gold robot.

The resulting subset is then ranked using another characteristic of shiny surfaces: the expected spatial relations between those color-sets (white highlights surrounded by light-gold spots, surrounded by medium-gold surfaces, which in turn are surrounded with dark-golden edges). We expressed this property using color correlograms, ranking the relevant transitions.

Using this elaborate approach, we managed to retrieve one of the correct answers, but no higher than position 30. We retrieve many 'golden' images with elements satisfying our limited definition of shininess (most of them not 'metallic'), but the properties of metal surfaces must be modeled more realistically to get more convincing results.

Topic 32: Helicopter

The helicopter topic provides three audio examples, and we experimented with the audio analogon of query articulation in an attempt to find scenes with helicopters. We hoped to specify the characteristics of a helicopter sound as a combination of two filters: (1) a repetitive pattern using periodicity of the audio spectrum, and (2) a concentration of energy in the lower frequencies, using spectral centroid and bandwidth features. Details of the techniques we tried can be found in [17].

Unfortunately, the helicopter sound in the known-item can only be noticed in the background, and some characteristics of the speech voice-over overlap with the idea of the second filter. It turns out the combination of filters can detect sounds corresponding to vehicles and airplanes, but we have not managed to tune the filters such that it singles out helicopters only.

4.3 Reflection

The highlighted known-item searches illustrate the idea underlying the process of query articulation, and demonstrate how query articulation may improve the results of multimedia retrieval dramatically. Without the elicitation of such relatively exact queries, none of these topics could be solved using our limited feature models. The query articulation process studied for topics 25 and 32 (and even for topic 8) suffered however from the risk of overemphasizing precision, sacrificing overall recall. Especially if the features available in the system do not correspond closely to the particular characteristics of the desired result set, the current system does not provide sufficient support to assess suitability of candidate strategies. But, also if appropriate features are available, the resulting query may 'overlook' other possibilities; for example, our strategy would not find the lunar rover if appearing in a lunar crater or in a hangar on earth (so there is no visible black sky).

5 Lazy Users

In our interactive experiments, we assumed a 'lazy user' model: users investing only limited effort to express their information need. Our users view 20 result summaries at a time, after which they choose whether to look at more results from the current strategy, or formulate a new strategy. They are not expected to investigate more than 100 result summaries in total. Lazy users identify result sets instead of correct answers, so our interactive results are not 100% precision.

The combination strategies used to construct run 5 consisted of:

- choose the run that looks best;
- concatenate or interleave top-N from various runs;
- continue with an automatic, seeded search strategy.

For example, the strategy for topic 24 (Lynn Bondurant) used a seeded search based on run 3, which

was interleaved with the results of run 4. Surprisingly, the run with speech transcripts only turns out better than the combined run, although not on all topics. It has proven difficult to combine results of multiple input runs effectively. While lack of time did also play a role (the combination strategies were not tried very systematically), the results for topics 54 and 59 demonstrate that a lazy user can, based on a visual impression of a result set, inadvertently decide to discard the better results (in both cases, run 3 was better but run 4 was chosen as best answer). Tool support for such a combination process seems a promising and worthwhile research direction.

6 Discussion

A major goal of having a video retrieval task at TREC-10 was to research a meta-question: investigate (experimentally, through a 'dry-run') *how* video retrieval systems should be evaluated. Working on the task, we identified three concerns with the current setup of the evaluation:

- the inhomogeneity of the topics;
- the low quality of the data;
- the evaluation measures used.

Candidate participants all contributed a small number of multimedia topics, the union of which formed the topic set. Partly as a result of the different angles from which the problem of video retrieval can be approached, the resulting topic set is very inhomogeneous. The topic text may describe the information need concisely, but can also provide a detailed elucidation; topics can test particular detectors, or request very high-level information; and some topic definitions are plainly confusing, like 'sailboat on the beach' which uses a yacht on the sea as image example[6]. Thus, each subtask consisted of a mix of (at least) three distinct classes of topics: detector-testers, precise known-item topics, and generic searches. This inhomogeneity causes two problems: it complicates query analysis for automatic systems, and makes comparison between runs difficult (a single good detector can easily dominate an overall score like average precision).

The low quality of the video data provided another unexpected challenge. It makes some topics more complex than they seemed at first sight (like 'Jupiter'). Also, the results obtained with the technique discussed in Section 2.2.5 are much lower than the application of the same paradigm on for example the Corel photo gallery. In fact, we observed that in many cases the color distributions to a large extent are a better indication of the similarity in age of the data than of the true video content. Of course, this can also be viewed as a feature of this data set rather than a concern. Experiments discussed by Hampapur in [5] showed as well how techniques behaving nicely on homogeneous, high quality data sets are of little value when applied to finding illegal copies of video footage on the web (recorded and digitized with widely varying equipment).

The third concern, about the evaluation measures, is based on two slightly distinct observations. First, our lazy user model returns shots as answers for known-item queries, but these are often shorter than 1/3 of the scenes that should be found. The chosen evaluation metric for known-item topics thus deems our answers not relevant, while this could be considered open for discussion: a user could easily rewind to the start of the scene.

Second, an experimental setup that solves the interactive topics by handpicking correct answers should probably result into 100% precision answer sets. First of all, this indicates that precision is not the right measure to evaluate the results of the interactive task. Lower scores on precision only indicate inter-assessor disagreement (viewing the user as just another assessor), instead of the precision of the result set. Another example of this phenomenon can be found in the judgments for topic 59 on runs 4 and 5, where identical results were judged differently.[7] The significant difference in measured performance indicate that the current topics and relevance judgments should probably not be used as ground truth data for laboratory experiments.

As a concluding remark, it is not so clear how realistic the task is. First of all, no participant seemed to know how to create 'doable' topics for the BBC data, while those video clips are drawn from a real video archive. Also, it seems unlikely that a user with state-of-the-art video retrieval tools could have beaten a naive user who simply scrolls through the relatively small set of keyframes. A larger collection would give video retrieval systems a fairer chance, but the engineering problems (and cost) arising might discourage participation in the task.

7 Conclusions

In spite of the issues raised in the discussion, we believe the TREC video evaluation is a strong initiative

[6] Shame on us – we contributed this topic ourselves.

[7] This may also have been a case of *intra*-assessor disagreement.

that was much needed to advance the field of multimedia retrieval, and it has already pointed us to a range of problems that we may never have thought of without participation.

Our evaluation demonstrates the importance of combining various techniques to analyze the multiple modalities. The optimal technique depends always on the query; both visual and speech can prove to be the key determining factor, while user interaction is crucial in most cases. The final experiment attempted to deploy all available information, and it seems worthwhile to investigate in research into better techniques to support choosing a good combination of approaches. In some cases, this choice can already be made automatically, as demonstrated in run 1; but, in cases like the known-item searches discussed for run 4, user interaction is still required to decide upon a good strategy.

Our (admittedly poor) results identify many issues for future research: new and improved detectors (better suited for low-quality data), better combination strategies, and more intelligent use of the user's knowledge. The integration of supervised and unsupervised techniques for query formulation form a particular research challenge.

Acknowledgments

Many thanks go to Alex Hauptman of Carnegie Mellon University for providing the output of the CMU large-vocabulary speech recognition system.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley, Wokingham, UK, 1999. 3

[2] P. Bosch, A. van Ballegooij, A.P. de Vries, and M.L. Kersten. Exact matching in image databases. In *Proceedings of the 2001 IEEE International Conference on Multi media and Expo (ICME2001)*, pages 513–516, Tokyo, Japan, August 22–25 2001. 6

[3] Arjen P. de Vries. The Mirror DBMS at TREC-9. In Voorhees and Harman [19], pages 171–177. 3

[4] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. Pattern Anal. Machine Intell.*, to appear, November, 2001. 3

[5] A. Hampapur and R. Bolle. Comparison of distance measures for video copy detection. In *Proceedings of the 2001 IEEE International Conference on Multi media and Expo (ICME2001)*, Tokyo, Japan, August 22–25 2001. 9

[6] Djoerd Hiemstra. *Using language models for information retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001. 3, 4, 5

[7] Djoerd Hiemstra and Wessel Kraaij. Twenty-One at TREC-7: Ad-hoc and cross-language track. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Seventh Text Retrieval Conference TREC-7*, number 500–242 in NIST Special publications, pages 227–238, 1999. 3

[8] J. Huang, S.R. Kumar, M. Mitra, W. Zhu, and R. Zahib. Spatial Color Indexing and Applications. *International journal of Computer Vision*, 35(3):245–268, 1999. 7

[9] Philippe Joly and Hae-Kwan Kim. Efficient automatic analysis of camera work and microsegmentation of video using spatiotemporal images. *Signal Processing: Image Communication*, 8(4):295–307, 1996. 2

[10] Wessel Kraaij and Thijs Westerveld. TNO/UT at TREC-9: How different are web documents? In Voorhees and Harman [19], pages 665–671. 3

[11] N.M. Laird, A.P. Dempster, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, series B*, 39(1):1–38, 1977. 5

[12] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 1998. 3

[13] A.W.M. Smeulders, S. Santini M. Worring, A. Gupta, and R. Jain. Content based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, Dec. 2000. 1, 2

[14] J.R. Smith and S.-F. Chang. VisualSEEk: a fully automated content-based image query system. In *ACM Multimedia 96*, Boston, MA, 1996. 6

[15] C.G.M. Snoek. Camera distance classification: Indexing video shots based on visual features. Master's thesis, Universiteit van Amsterdam, October 2000. 3

[16] TextBridge SDK 4.5. http://www.scansoft.com. 3

[17] Alex van Ballegooij, Johan List, and Arjen P. de Vries. Participating in Video-TREC with Monet. Technical report, CWI, 2001. 6, 8

[18] N. Vasconcelos and A. Lippman. Embedded mixture modelling for efficient probabilistic content-based indexing and retrieval. In *Multimedia Storage and Archiving Systems III*, volume 3527 of *Proceedings of the SPIE*, pages 134–143, 1998. 3, 4, 5

[19] E.M. Voorhees and D.K. Harman, editors. *Proceedings of the Ninth Text Retrieval Conference (TREC-9)*, number 500–249 in NIST Special publications, 2001. 10

[20] M. Worring and L. Todoran. Segmentation of color documents by line oriented clustering using spatial information. In *International Conference on Document Analysis and Recognition ICDAR'99*, pages 67–70, Bangalore, India, 1999. 3