Using Video Objects and Relevance Feedback in Video Retrieval

Sorin Sav[a], Hyowon Lee[b], Alan F. Smeaton[b], Noel E. O'Connor[b] and Noel Murphy[a]

[a]Centre for Digital Video Processing, Dublin City University, Glasnevin, Dublin 9, Ireland;
[b]Centre for Digital Video Processing and Adaptive Information Cluster, Dublin City University, Glasnevin, Dublin 9, Ireland;

ABSTRACT

Video retrieval is mostly based on using text from dialogue and this remains the most significant component, despite progress in other aspects. One problem with this is when a searcher wants to locate video based on what is appearing in the video rather than what is being spoken about. Alternatives such as automatically-detected features and image-based keyframe matching can be used, though these still need further improvement in quality.

One other modality for video retrieval is based on segmenting objects from video and allowing endusers to use these as part of querying. This uses similarity between query objects and objects from video, and in theory allows retrieval based on what is actually appearing on-screen. The main hurdles to greater use of this are the overhead of object segmentation on large amounts of video and the issue of whether we can actually achieve effective object-based retrieval.

We describe a system to support object-based video retrieval where a user selects example video objects as part of the query. During a search a user builds up a set of these which are matched against objects previously segmented from a video library. This match is based on MPEG-7 Dominant Colour, Shape Compaction and Texture Browsing descriptors. We use a user-driven semi-automated segmentation process to segment the video archive which is very accurate and is faster than conventional video annotation.

Keywords: Video retrieval, shot retrieval, object segmentation, object retrieval

1. INTRODUCTION

In this paper we are concerned with the interactive retrieval of short segments of video commonly referred to as shots, from a potentially large video archive. This is in contrast to the most often used scenario for video searching where we are searching for entire video programmes such as when searching a Personal Video Recorder (PVR) or movie library. Indexing of entire video programs, such as movies, entire TV news broadcasts or TV programs, is easily done using metadata and there are many example systems which support access to video in such a way. Here we are concerned with direct access to video content, based on what is actually appearing in the video frame or from the associated audio track. Scenarios where such retrieval is needed includes trying to locate specific scenes in a movie, such as when Tom Cruise used the gesture-based video search tool in the movie *Minority Report* or when the alien creature bursts out of the stomach of John Hurt in the movie *Alien*. Even when searching an archive of personal video content (from a camcorder, for example), we usually want to locate individual shots, such as the shot of your son blowing out the candles on his 4th birthday or the shot of your children playing on the beach on your vacation last year.

For the most part, the dominant approach to video searching, at the sub-video or "shot" level is based on text from automatic speech recognition (ASR), from video OCR and from closed captions. This works well when the text content is in some way descriptive of the visual content, or at least the on-screen video is illustrative of the text content. This occurs in video genres like broadcast news or TV documentaries, but not in movies, CCTV, home movies, or other kinds of TV program. When text-based video search doesn't work, or even when it does, then a second type of video search that is useful is matching images which illustrate the information need against keyframes. Such matching is usually based on visual similarity between images and will be useful when

Send correspondence to Alan Smeaton, Alan.Smeaton@DCU.ie

a searcher has a candidate query image, or can locate a keyframe serving this purpose from the video archive. A third common approach to video shot retrieval involves the definition of semantic concepts whose occurrences can be automatically determined and then used as filters or search criteria if the semantic annotations are reliable and accurate. This latter is what is used in manual video annotation in TV archives and elsewhere but it is manual and expensive to construct and maintain, and there is much work and interest to automate this.

A hugely important component of video navigation where a user locates a video clip which is interesting or even relevant and wants to find more clips like that one, is *browsing*. One reason for this is that its difficult for users to specify queries to video retrieval systems because the user has to second-guess what may be in the ASR text, or guess which features may or may not be present, but once you find a good video clip it is easy to say *"find me more like that"*. All these approaches to video shot search can be demonstrated by several research groups working on collections of video which are large by personal standards but small by the standards of archives and broadcasters.

The information needs users have when searching video can vary from broad overview-type searching such as searching for any kinds of video clips of a professional bicycle race, to specific instance searching such as searching for a clip of the Tour de France yellow jersey race-leader ascending l'Alpe d'Huez. Video queries such as these can use the text search by searching for *"tour de France"* or *"Lance Armstrong"* and expect that such words will appear in the video dialogue. When one instance of such a video clip is located a user might use keyframes from that clip as the basis for a search for more clips which have blue sky, rocky backgrounds, a road surface and a number of brightly-coloured blobs stretched horizontally across the frame, corresponding to cyclists. However since keyframe matching is done on the basis of the entire frame and uses such characteristics as colour and texture, such keyframes might also retrieve clips of brightly coloured flowers or motor cars against similar backgrounds. A user might also use semantic annotations which describe the keyframe/shot, in order to filter the dataset and remove shots which are not likely to be relevant to the query. Features such as *"outdoors"*, *"contains people"*, and *"road"* will eliminate many non-relevant shots but unless the size of the ontology of such features is very large and each feature is automatically detected with a high degree of accuracy, neither of which is true at the moment, semantic annotations can serve only to partially reduce the set of video clips to be browsed. In practice, in contemporary video retrieval systems, a combination of text search, keyframe matching, and feature annotations are often used together and provide the most useful way to search video when operating *collectively*.

One feature of video that could potentially add a lot to the quality of video retrieval and which we have not mentioned yet is to use the occurrence of *objects* in the video, as part of the query. For example, if we were able to automatically detect the occurrence of a bicycle on a video clip then, in combination with the other strategies mentioned above, we could eliminate coloured flowers and cars from our l'Alpe d'Huez professional cyclist query. Not only that but once we've located some clips containing bicycles then we could add the actual instances of bicycles, isolated from their backgrounds, to the query. This is not the same as detecting the occurrence of bicycles as a general feature of video but involves actually identifying the specific occurrence of bicycle(s) in the video and then being able to do something useful with those occurrences, such as use them in querying.

In this paper we concentrate on video object retrieval, using video objects as the query, and we examine how achievable such retrieval actually is given current techniques. We do this by building a standalone system which supports retrieval of video based on video object occurrences. In practice such a system would not be very useful on its own but would form one component of an overall video retrieval system.

The rest of this paper is organised as follows. In the next section we introduce the task of object segmentation in video and in images and we give a brief overview of other approaches to video object retrieval which defines the state of the art in this area. Section 3 describes the system we have built and focuses on the way in which objects are segmented and matched. In section 4 we examine the concept of query splitting which we support and which addresses the situation where a user's example objects are so numerous that they add noise to the query and actually reflect 2 or more sub-queries which should be split and executed separately. Section 5 gives two worked scenarios which illustrate object-based video retrieval and the way queries can be automatically partitioned. A concluding section completes the paper by outlining planned future work.

2. VIDEO RETRIEVAL USING VIDEO OBJECTS: RELATED WORK

Even though video object segmentation is a very challenging task and is far from completely solved, there are some examples of work which supports video retrieval based on video objects and we briefly summarise some of these here.

In[1] the approach is to segment whole images into regions using visual characteristics, and then refer to a set of homogeneous regions as an object, though these are not objects in the semantic sense as we mean in this paper. An "object" retrieval system based on this approach is evaluated on a short animated cartoon rather than on natural video content, with a ground truth of object occurrences and in[1] they demonstrate the accuracy of an approach to locating objects in animated cartoon video based on a user query specified as a set of regions. Similarly in[2] there is another proposal for locating arbitrary-shaped objects without them being defined as real objects, this time based on shapes and shape deformations over time, with another set of evaluation figures on measuring the accuracy of locating query objects in video sequences as measured against humans locating the same objects. 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.

Sivic and his colleagues take an approach whereby the user is asked to perform object segmentation in the query keyframe and this object is then matched and highlighted against similar objects appearing in shots. The approach uses contiguous frames within a shot to improve the estimation of object occurrences by addresses changes in the appearance of an object due to change in camera viewpoint, in illumination, in object occlusion or in object movement. The approach thus goes much further than using jost keyframes, as most other work does, and is illustrated working on the movie *Groundhog Day* in[3] with a more detailed presentation of the image processing in.[4]

Work reported in[5] addresses object segmentation and retrieval based on a complex approach to motion representation and concentrates on the object tracking without actually segmenting the semantic object. This rather neatly avoids the problem of having to segment objects. The paper reports some preliminary experiments where similar objects which have a similar trajectory to the query clip and appear in similar video compositions, can be located, though a thorough evaluation is needed. Similar work, also operating on video rather than video keyframes, is reported in[6] where they automatically segment video frames into regions based on colour and texture, and then track the largest of these through a video sequence. Like the work of Liu et al in[5] they do not operate on segmented video objects but more like video *blobs*. In this approach a user query is not a segmented object but an object appearing in a query video clip. The work reported in[6] will search using a query video clip to find video sequences similar in terms of object motion, as well as edges, texture, and colours and this has been tested on a corpus of natural video.

While most of the work mentioned above is quite recent and suggests that object-based video retrieval is a new development, this is not true, with work on video retrieval using objects being reported more than 10 years ago, e.g..[7] Clearly the notion of using video objects for retrieval has been desirable for some time, but only very recently has technology started to allow even very basic object-location functions on video.

3. OBJECT MODELING FOR ITERATIVE REFINEMENT

When textual annotation is not available for image retrieval, a description of the targeted concept for retrieval can be difficult for the user. One popular mechanism for query initialization in image retrieval in such scenarios is query-by-example (QBE). The QBE strategy offers an elegant and compact solution to query formulation and we use this approach in the video object-based retrieval which we illustrate in this paper. In this section we shall describe the system in more detail.

3.1. System Description

The system's main capability which we illustrate in this paper is to iteratively improve the object-object similarity measure in the database using a user's query formulations during a search session to dynamically model objects in the database and to retrieve objects according to the formed model.

Figure 1 shows an overview of the system off-line indexing and on-line retrieval processes. Our system processes one object from each keyframe taken from each shot in the video and stores these in the database to be

used in the retrieval process during an interactive search session (see top left of Figure 1). For each keyframe the most representative video objects are extracted using the interactive tool for segmentation described in Section 3.2. We use keyframes automatically extracted from the TRECVid 2003[8] test corpus, as well as images from the well known Corel test corpus.

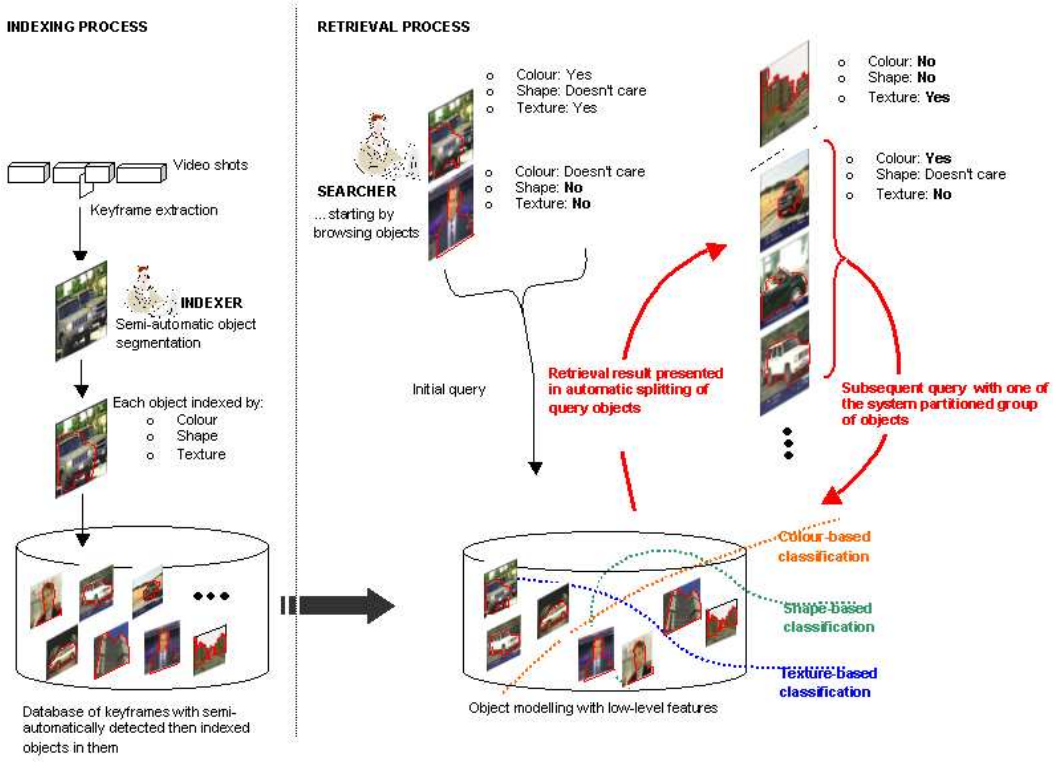


Figure 1. System Overview showing off-line indexing and interactive searching

Once segmented, each object is automatically indexed by colour, shape and texture using the MPEG-7 Dominant Colour, Shape Compaction and Texture Browsing descriptors detailed in Section 3.3. This completes the offline object segmentation and indexing process (left side of Figure 1). Determining similarity among objects for retrieval purposes is done during interactive search as the system receives more information from the user dynamically. A user starts with an initial query examples to search for similar objects in the system corpus, from which the system starts clustering the images based on this initial query (see middle of Figure 1). Retrieved results and the subsequent queries by the user are iterated and as this process continues the internal clustering becomes more accurate. The system also features a mechanism where this accuracy can be maintained even when the user adds inconsistent examples for querying (query split feature) which will be detalied in Section 4

3.2. Video Object Segmentation

Since reliable unsupervised general-purpose image segmentation is considered as generally unattainable, in this approach user interaction is performed in order to define what objects are to be segmented within an image.[9] It is desirable that user interaction be easily and quickly performed. To this end, we allow the user to mark objects to be segmented via a simple mouse drag over each object. The interface provided and an example of user interaction is presented in Figure 2. The Recursive Shortest Spanning Tree (RSST) algorithm, described in[10] is used initially to automatically pre-segmented the image into a large number of uniform colour regions. The result of user interaction is then used in an fully automatic region assignment process. Regions coincident with an object's mouse drag are added to that object. Unclassified regions are assigned to competing objects using a normalized distance criterion similar to that used in the RSST algorithm. For complex objects, user

interaction (and the subsequent automatic process) can be iteratively applied in order to refine the segmentation result.

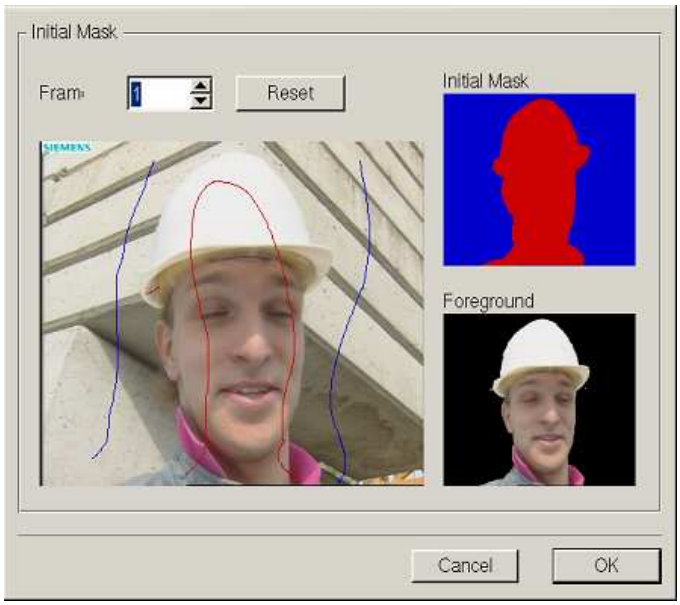


Figure 2. Video object segmentation via user interaction

3.3. Feature Descriptors For Objects

The features we selected for image representation are colour, shape and texture as they are directly related to human perception and independent of each other. In our system the features describe only the image foreground, i.e. the segmented object.

3.3.1. Colour Representation

To represent colour we adopted the MPEG-7 Dominant Colour Descriptor (DCD),[11] which is used by many retrieval systems. The recommended distance to be used with DCD is[12] :

$$D_{DCD}(Q,I) = \left(\sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2\alpha_{1i,2j} p_{1i} p_{2j} \right)^{1/2} \tag{1}$$

where N is a set of colour vectors c_i, and p_i their percentages. The similarity coefficient $\alpha_{k,l}$ between two RGB color vectors c_k and c_l is calculated as:

$$\alpha_{k,l} = \begin{cases} 1 - \frac{D_{k,l}}{D_{max}}, & D_{k,l} \leq T_d \\ 0, & D_{k,l} > T_d \end{cases} \tag{2}$$

In expression 2 $D_{k,l} = \| c_k - c_l \|$ represents the Euclidian distance between two colour vectors. $T_d = 20$, $\alpha = 1$, and $D_{max} = \alpha T_d = 20$, follow the values given in.[13]

3.3.2. Shape Representation

Shape description and similarity is an extremely complex research topic. The 2D projection on the image plane, elastic deformations of the object, and diversity of shapes in which instances of the same semantic object appear in the real world are common problems that must be considered for shape similarity. In our work, we use a relatively simple shape descriptor corresponding to the compactness moment γ,[14] defined by Equation 3. This is a simple and robust descriptor that can indicate a degree of shape similarity.

$$\gamma = \frac{P_2}{4\pi A} \tag{3}$$

where A is the area and P perimeter of the video object defined as:

$$P = \sum_{i=1}^{N-1} \parallel x_{i+1} - x_i \parallel + \parallel x_N - x_1 \parallel \tag{4}$$

3.3.3. Texture Representation

In our system texture is represented with the MPEG-7 Texture Browsing Descriptor.[11] This descriptor is expressed as a set of 24 Gabor wavelets[15] $g_{m,n}(x,y)$ (6 orientations, 4 scales) obtained by appropriate rotations and dilations of the a two dimensional Gabor function:

$$\begin{array}{l} g_{m,n}(x,y) = a^{-m}G(x',y'), \qquad a > 1 \\ x' = a^{-m}(x\cos\theta + y\sin\theta) \\ y' = a^{-m}(-x\sin\theta + y\cos\theta) \end{array} \tag{5}$$

where $\theta = n\pi/K, K$ is the total number of orientations and a^{-m} is the scale factor. $G(x',y')$ is the Fourier transform of a two dimensional Gabor $g(x,y)$ function:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jW\right] \tag{6}$$

Given an image $I(x,y)$ its Gabor wavelet transform is then defined as:

$$W_{m,n}(x,y) = \int\int I(x',y')g_{m,n} * (x - x', y - y')dxdy \tag{7}$$

where * indicates the complex conjugate and $g_{m,n}$ are the Gabor wavelets. It is assumed that the local texture regions are spatially homogeneous, and the mean $\mu_{m,n}$ and the standard deviation $\sigma_{m,n}$ of the magnitude of the transform coefficients are used to represent the region classification for retrieval purposes[15]:

$$\begin{array}{l} \mu_{m,n} = \int\int |W_{m,n}(x,y)|dxdy \\ \sigma_{m,n} = \sqrt{\int\int (|W_{m,n}(x,y)| - \mu_{m,n})^2 dxdy} \end{array} \tag{8}$$

The resulting vector has $\mu_{m,n}$, $\sigma_{m,n}$ feature components. Then the distance between two patterns i and j in the texture space[15] is defined as:

$$\begin{array}{l} d(i,j) = \sum_m \sum_n d_{m,n}(i,j) \\ d_{m,n}(i,j) = |\frac{\mu_{m,n}^{(i)} - \mu_{m,n}^{(j)}}{\alpha(\mu_{m,n})}| + |\frac{\sigma_{m,n}^{(i)} - \sigma_{m,n}^{(j)}}{\alpha(\sigma_{m,n})}| \end{array} \tag{9}$$

where $\alpha(\mu_{m,n})$ and $\sigma(\mu_{m,n})$ are the standard deviations of the respective features over the entire collection and are used to normalise the individual feature components.

4. VIDEO OBJECT RETRIEVAL AND AUTOMATIC QUERY SPLITTING

Our system makes use of implicit explanations by visually showing the query documents (video objects) grouped in clusters based on their feature similarity. This visual representation provides the user with an intuitive explanation regarding the distribution of the relevant documents in the searched collection. To build a query the user can indicate positive and negative examples of video objects. By grouping the query objects into clusters, the system is suggesting to the user that their information need has actually diversified into two or more distinct categories of object retrieval which has already been differentiated by the system. This reflects the case of a user wishing to explore two aspects or branches of their query, which our system can support as we show later, and this neatly maps onto the ostensive model of retrieval[16] where a user is encouraged to explore one aspect freely until it is exhausted and then return to this point and launch an exploration into the second aspect.

Once the user had provided (through relevance feedback) a set of objects as an indication of the objects they wish to retrieve, these are analysed in terms of colour, shape and texture. Considering these features as independent of each other we define an object-to-object similarity measure S_{object} as:

$$S_{object}(i,j) = \alpha S_{colour}(i,j) + \beta S_{shape}(i,j) + \gamma S_{texture}(i,j) \tag{10}$$

where α, β and γ are normalisation factors for the colour S_{colour}, shape S_{shape} and texture $S_{texture}$ similarity measures. For each feature the corresponding similarity measure is independently computed and adjusted to better match the positive examples provided by the user through relevance feedback. The α, β and γ factors are directly proportional to the number of positive examples provided by the user for each of the respective features.

We assume that the positive examples can be modelled by a Gaussian mixture model where mixture component is a Gaussian with mean μ and covariance matrix Σ:

$$p(\varepsilon|j) = \frac{1}{2\pi|\Sigma_j|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\varepsilon-\mu_j)^T\Sigma_j^{-1}(\varepsilon-\mu_j)} \tag{11}$$

The mean μ and variance Σ are estimated from examples labelled as relevant (positive). The density probability functions of the Gaussian mixture are estimated through the Estimation-Maximisation (EM)[17] procedure.

At this point we obtain a vector of parameters (μ,Σ) for the Gaussian mixture that models each feature (colour, shape, texture).

$$\begin{aligned}
\overline{v}_{colour} &= ((\mu^{(1)}_{colour}, \Sigma^{(1)}_{colour}),..,(\mu^{(n)}_{colour}, \Sigma^{(n)}_{colour})) \\
\overline{v}_{shape} &= ((\mu^{(1)}_{shape}, \Sigma^{(1)}_{shape}),..,(\mu^{(m)}_{shape}, \Sigma^{(m)}_{shape})) \\
\overline{v}_{texture} &= ((\mu^{(1)}_{texture}, \Sigma^{(1)}_{texture}),..,(\mu^{(p)}_{texture}, \Sigma^{(p)}_{texture}))
\end{aligned} \tag{12}$$

The components of these vectors are then combined such that each component of the colour vector is grouped with each component of the shape and texture vectors, constructing a query triplet.

$$query_{(i,j,k)} = ((\mu^{(i)}_{colour}, \Sigma^{(i)}_{colour})(\mu^{(j)}_{shape}, \Sigma^{(j)}_{shape})(\mu^{(k)}_{texture}, \Sigma^{(k)}_{texture})) \tag{13}$$

Each query triplet is a possible search direction (sub-query). However there is a possibility of the number of queries growing exponentially with the number of feature's components therefore we need to constrain the expansion of the query triplets by merging together close related triplets. The final purpose is to obtain a small number of mutually complementary sub-queries which represent clusters of objects (sub-queries) where the objects included in one cluster do not exist in any other cluster (sub-query). The merging procedure uses the Mahalanobis distance[18] to compute the distances between each two pairs of feature's triplets. The Mahalanobis distance is expressed as:

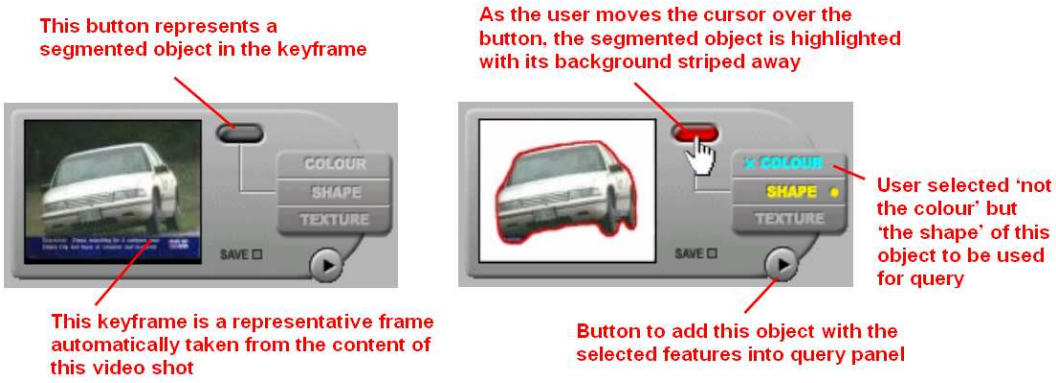


Figure 3. A shot representation and the interaction with an object for query specification

$$r^2 = (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \ . \tag{14}$$

where x is the colour, shape and texture triplet, μ_i is the mean of the resulting triplet, and Σ_i^{-1} is the diagonal covariance matrix for each of the colour, shape and texture triplets. At each merge iteration the pair of triplets having the smallest Mahalanobis distances are merged. The merging is repeated until all sub-queries (triplets) are mutually complementary.

The video objects associated with the feature's triplets resulted from the merging step the are then presented to the user. The user has the option to select one of the displayed queries (group of objects) as the active query in the next iteration. In one sense what we have done is to automatically split the initial group of query objects in sub-queries where each individual sub-query could represent a set of objects which are similar to each other but dissimilar to rest of the query objects.

In the next retrieval step we calculate the similarity distance from the mean μ and variance Σ of each feature in the active query to the features of the objects in the collection. The objects that have the highest similarity score are retrieved and presented to the user. The estimation-maximisation and query construction steps are repeated when new examples are labelled by the user.

5. RETRIEVAL SCENARIOS FOR VIDEO OBJECT RETRIEVAL

In this section we illustrate how object segmentation, user interaction with the segmented objects, relevance feedback with objects, object classification and the automatic query splitting mechanism explained in the previous section have all been combined to work together into one operating system.

Figure 3 illustrates a shot representation and how the user can interact with it to see the segmented object within a keyframe and to specify the features (colour, shape and texture) of that object that they wish to use in retrieval. The shot representation and the interaction illustrated in Figure 3 is the basic mechanism to allow the user to interact with objects segmented by the system, and to enable object-based actions such as viewing, selecting, saving, and using an object as an example query. Figure 4 is a screen shot of the overall interface to the system which incorporates this interaction mechanism. A user starts by browsing some example keyframes on the left-most column. For each keyframe, the user can check exactly what part in the keyframe has been segmented as an object by bringing the mouse cursor over the button beside the keyframe that represents an object. When she finds an object that is similar to her needs, she specifies values for the three features (colour, shape and texture) as Positive or Negative or Neutral and then clicks on the round arrow button below. This copies the selected object and its feature specification into the query panel (the 2nd column in Figure 4). The user can remove the added object from the query panel or add as many objects as she desires. Clicking on the "FIND" button on the query panel triggers retrieval from the object database using the objects and its features collected on the query panel, and the result is displayed on the SEARCH RESULT column (3rd column in Figure 4). From the search result the user can further browse objects and add objects into the query panel. At any

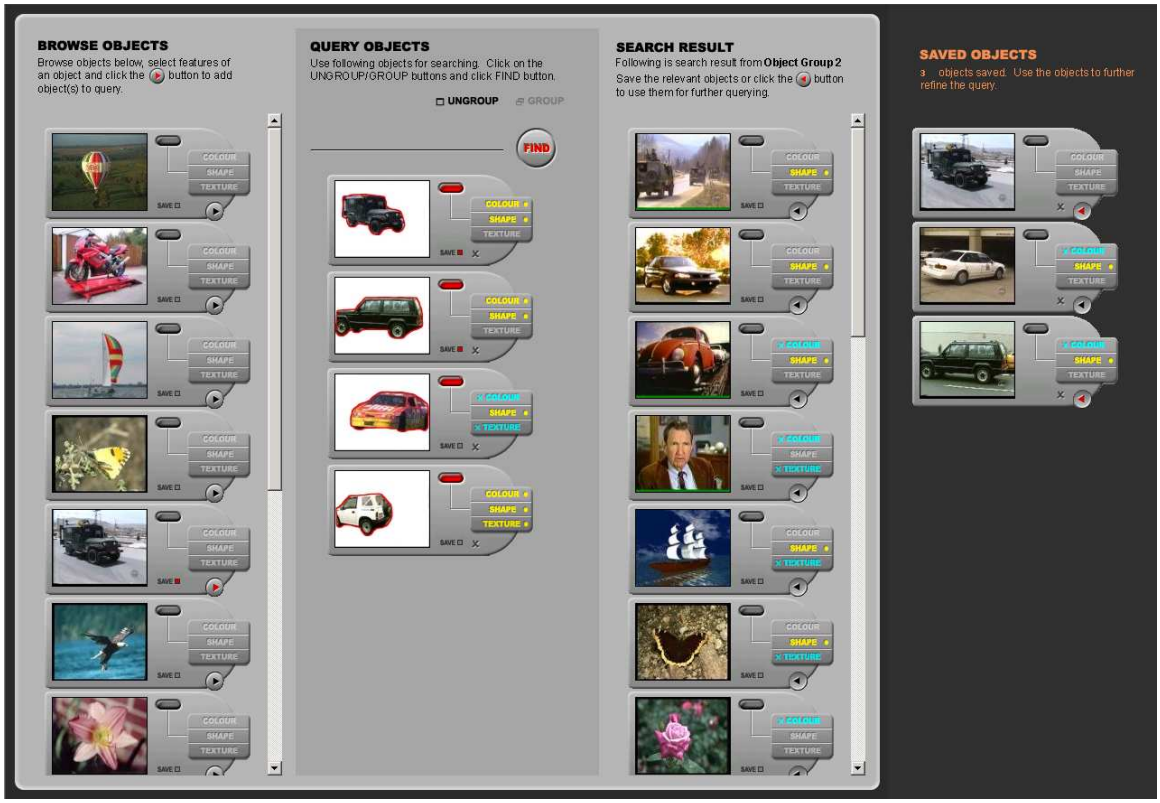
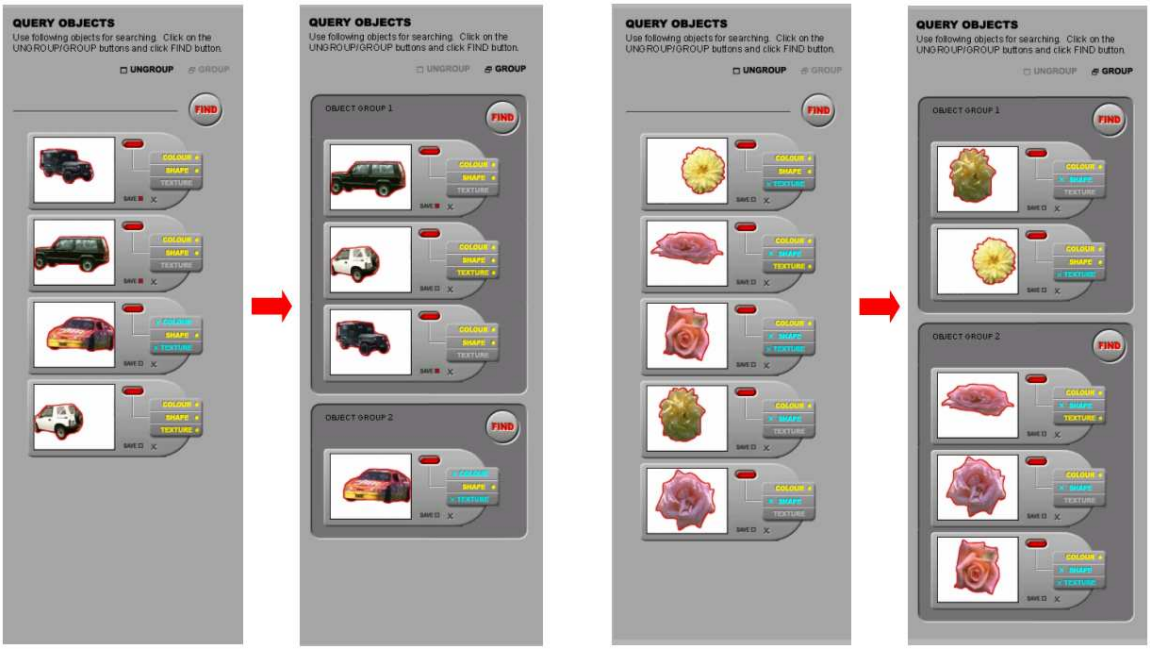


Figure 4. Overall interface of the system

point of browsing, clicking on the "SAVE" button will copy the entire shot into the SAVED OBJECTS area (the last column in Figure 4) - this is the area the user can save objects for whatever use at hand, and can be considered as a "bookmark" or "my favourites". Figure 4 shows the screen after the user has gone through 3 iterations of searching, currently having added four car objects to the query panel, and found three relevant cars which she saved on the SAVED OBJECTS column.

As more and more objects are added to the query panel, the system's underlying classification of objects becomes more and more refined. However, as mentioned in Section 3, retrieval performance depends on weather a "good" set of example objects has been collected in the query panel: if very different objects with conflicting features have been specified in the query panel, the system's classification will become more confused than refined. The automatic object grouping feature of the system is an important search support component of the system to help the user be aware of this possible discrepancy in the objects she has added to the query panel. The four car objects in the query panel in Figure 4, although to us are seemingly reasonably and look similar to each other, may have some discrepancy among themselves and thus not have an overall combined positive effect in refining the classification. Thus, instead of initiating a query based on all four car objects, the system can split the four cars into groups based on the internal classification that can help better refinement when queried separately.

Figure 5 illustrates this in two different cases, showing the partial screen capture of the query panel (2nd column in Figure 4). In Figure 5 (a), the query panel part of Figure 4 is illustrated. The user has added four car objects thinking they will all help her search, but not all of these four will be conducive to the accurate classification of the objects in the database: the user clicks on the "GROUP" button, and the system splits the added objects using its internal clustering mechanism as described in Section 3. In this example, three of the added cars are square-shaped and one is more round shape and red colour, thus the system automatically grouped them separately. Now the user realises the difference between the objects she has been adding, and focuses her



(a) Four car objects split into two groups: three square-shaped cars and one round, red car. The user can now use either group for querying separately.

(b) Five flower objects split into two groups: two yellow flowers and three pink flowers. The user can now focus on either group of objects.

Figure 5. Two example cases of automatic splitting of the relevance feedback objects

search using either of the groups by clicking on the "FIND" button within a group. Adding more objects in the split query panel will automatically put in one of the groups if it helps the refinement of classification, or form a new group if it cannot find an appropriate group.

In Figure 5 (b), another example is illustrated showing the user looking for flowers, mainly by its colour. She has currently added five example flowers into the query panel, without realising that not all of these five are similar to each other and thus will not be helpful in more accurate retrieval. When she clicks on "GROUP'' button, the system splits the flowers into two groups - one with two yellow flowers and one with three pink flowers. Now the user notices that she has been adding two different types of flowers and thus will continue her search by focusing on either the yellow or pink groups of flowers. These examples show how the system helps the user's relevance feedback process by automatically splitting the feedback objects into similar groups to guide the user into more specific and accurate searching.

6. CONCLUSION

This paper describe a system to support object-based video retrieval where a user selects example video objects as part of the query. During a search a user builds up a set of these which are matched against objects previously segmented from a video library. This match is based on MPEG-7 Dominant Colour, Shape Compaction and Texture Browsing descriptors. We use a user-driven semi-automated segmentation process to segment the video archive which is very accurate and is faster than conventional video annotation.

Our system makes use of implicit explanations by visually showing the query documents (video objects) grouped in clusters based on their feature similarity. This visual representation provides the user with a intuitive explanation regarding the distribution of the relevant documents in the searched collection.

In its present form our system may not to be suitable for an operational context, but the point of developing it was to demonstrate how an object-based query formulation mechanism could be realised to help dynamically refine the object model in the database and enhance retrieval.

For future work we will focus on segmenting more than one object from each keyframe. Our user interface accommodates interaction with more than one object in a single keyframe (by way of multiple buttons). Currently a keyframe from a shot is used to segment objects however a more complete solution would be to use all frames within the shot, which could further provide additional information on the object from its movement and trajectory rather than from just the keyframe. Additionally we will incorporate object-based retrieval into some evaluation framework like TRECVid as one component of a video retrieval system where we can thoroughly measure its true contribution to video retrieval.

ACKNOWLEDGMENTS

This work is partly supported by Science Foundation Ireland under grant 03/IN.3/I361. The support of Enterprise Ireland is also gratefully acknowledged.

REFERENCES

1. L. Hohl, F. Souvannavong, B. Merialdo, and B. Huet, "Enhancing latent semantic analysis video object retrieval with structural information," *Proceedings of the International Conference on Image Processing (ICIP 2004)* **Vol. 3**, pp. 1609–1612, Singapore, October, 2004.
2. B. Erol and F. Kossentini, "Shape-based retrieval of video objects," *IEEE Transactions on Multimedia* **7(1)**, pp. 179–182, 2005.
3. J. Sivic, F. Shaffalitzky, and A. Zisserman, "Efficient object retrieval from videos," *Proceedings of the 12th European Signal Processing Conference (EUSIPCO 2004)* , Vienna, Austria September 2004.
4. J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV 2003)* , Nice, France, 2003.
5. C.-B. Liu and N. Ahuja, "Motion based retrieval of dynamic objects in videos," *Proceedings of the 12th annual ACM International Conference on Multimedia (MM 2004)* , pp. 288–291, New York, NY, USA, 2004.
6. M. Smith and A. Khotanzad, "An object-based approach for digital video retrieval," *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2004)* **Vol. 1**, pp. 456–459, Las Vegas, NV, USA, April 2004.
7. E. Oomoto and K. Tanaka, "Ovid: Design and implementation of a video-object database system," *IEEE Transactions on Knowledge and Data Engineering* **5(4)**, pp. 629–643, 1993.
8. TRECVid 2003, http://www-nlpir.nist.gov/projects/trecvid.
9. N. O'Connor, T. Adamek, S. Sav, N. Murphy, and S. Marlow, "Qimera: a software platform for video object segmentation and tracking," *Proceedings of the International Workshop on Image Analysis For Multimedia Interactive Services(WIAMIS 2003)* , pp. 204–209, , London, April 2003.
10. O. Morris, M. Lee, and A. Constandinides, "Graph theory for image analysis: an approach based on the shortest spanning tree," *Proceedings of the IEE* **Vol. 133**, pp. 146–152, April 1986.
11. P. Salambier and J. Smith, "Mpeg-7 multimedia descriptions schemes," *IEEE Transactions on Circuits and Systems for Video Technology* **11**, pp. 748–759, June 2001.
12. B. Manjunath, P. Salambier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, Wiley, New York, USA, 2002.
13. A. Kushki, P. Androutsos, K. Plataniotis, and A. Venetsanopoulos, "Query feedback for interactive image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology* **14(5)**, pp. 644–655, May 2004.
14. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.
15. P. Salambier and J. Smith, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18(8)**, pp. 837–842, August 1996.
16. I. Campbell and C. van Rijsbergen, "Texture features for browsing and retrieval of image data," *Proceedings of the 2nd International Conference on Conceptions of Library and Information Science (CoLIS 2)* , Copenhagen, Denmark 1996.
17. T. K. Moon, "The expectation-maximisation algoritm," *IEEE Signal Processing Magazine* , pp. 47–60, November 1996.

18. F. Fessant, P. Aknin, L. Oukhellou, and S. Midenet, "Comparison of supervised self-organizing maps using euclidian or mahalanobis distance in classification context," *Proceedings of the 6th International Work Conference on Artificial and Natural Neural Networks (IWANN2001)* , Granada, Spain 2001.