Using Segmented Objects in Ostensive Video Shot Retrieval

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

Centre for Digital Video Processing
Dublin City University
Glasnevin, Dublin 9, Ireland.
sorinsav@eeng.dcu.ie, hlee@computing.dcu.ie,
asmeaton@computing.dcu.ie, oconnorn@eeng.dcu.ie

Abstract. This paper presents a system for video shot retrieval in which shots are retrieved based on matching video objects using a combination of colour, shape and texture. Rather than matching on individual objects, our system supports sets of query objects which in total reflect the user's object-based information need. Our work also adapts to a shifting user information need by initiating the partitioning of a user's search into two or more distinct search threads, which can be followed by the user in sequence. This is an automatic process which maps neatly to the ostensive model for information retrieval in that it allows a user to place a virtual checkpoint on their search, explore one thread or aspect of their information need and then return to that checkpoint to then explore an alternative thread. Our system is fully functional and operational and in this paper we illustrate several design decisions we have made in building it.

1 Introduction and Background

The continuous expansion of video archives has resulted in an increasing demand for effective information management systems. Browsing keyframes or fast-forward or automatic summarisation are all useful tools for navigating small amounts but large volume search capabilities are also required as archive sizes grow. Current approaches to searching use either (1) the ASR text or OCR text from still frames, or (2) match an external image or an existing keyframe against shot keyframes using low-level features like colour histograms or texture or edges or (3) automatically assign concept features such as indoor, outdoor, faces, dialogue, building, landscape, camera zooming, etc. and use these to filter shots for subsequent browsing. Each have advantages and each can successfully address some types of video retrieval [1]. However, often our information need when we search is for a specific object such as a search for a shot containing a motorbike or car, or a shot containing a horse. In such cases the spoken dialogue (ASR) may not describe what is on camera, overall content in a sample keyframe or image may be totally different to one containing our target object and thus overall colour, texture, edges etc. will be different, and there may not be an available

concept feature detector for the object we are seeking. In such cases we need to search for actual objects, which is the focus of this paper.

Of course object search is not a panacea and works best when used as one of an available set of search tools including text, image match and feature filtering, all combined with a usable video browsing interface, but in this paper we concentrate on object retrieval as the retrieval tool. We have built a retrieval tool for video shot retrieval which retrieves based on objects and not based on the other modalities. The purpose here is not to do simple matching of an object from a query image against objects from a video keyframe but to use the selection of a *set of objects* in a query as the basis for retrieval. The purpose of our research is to explore object-based shot retrieval more than just object matching and as we will see this demands the formation and use of sets of query objects.

In the task of automatically segmenting and indexing objects in image/video content, the main difficulty is the diverse manifestation of an object in the image/video regardless of the object's inherent visual features such as colour, shape and texture. Due to factors such as different lighting conditions, different angles taken by the camera, and the degree and types of occlusions that often occur on objects, this makes the actual segmentation of an object as well as labelling the segmented object, for example a car, extremely difficult. This same problem of diverse manifestations of an object also occurs when a searcher has to give examples of an object during query formulation.

With this problem as the central issue, one workaround solution we have been exploring is to use ostensive relevance feedback, which takes a human user's judgements on object definitions, in retrieving objects. There is a long history of experimentation and successful use of relevance feedback in text-based information retrieval. This has included short-term modelling of a user's information need by dynamically updating the user's query formulation in mod-search as well as long-term modelling of user's needs by profiling his/her interests over time.

In this paper, we present an interactive, object-based search system that uses a novel, adaptive query formulation mechanism. As query formulation is the key element for getting feedback from the user in our approach, the system we have built incorporates a user interaction strategy in which a user can interact with segmented objects by way of highlighting them, selecting them, and then using them in subsequent query formulation.

The novelty of our work lies in using automatic query branching into an ostensive relevance feedback framework as a means to provide the user with knowledge about the distribution of object features in the video collection. This is a two way feedback where the system is instructed about the relevance of retrieved objects and the user receives explicit indications about the mapping of the query into the feature space. By being aware of the ramifications which a query has on the collection space, the user can better adapt the query and their feedback to more accurately select query objects relative to their information need.

The paper is organised as follows: in Section 2 we give an overview of ostensive relevance feedback as used for video object shot retrieval, Section 3 describe the algorithm used for video object retrieval and Section 4 presents the feature descriptors for video objects used in our system. The design of the object-user interaction mechanism and the front-end user interface is described in Section 5. Section 6 concludes the paper with our plans for extending the system's capability and further refining the user-interface.

2 Ostensive Relevance Feedback Applied to Video Shot Retrieval

The process of information retrieval is an inherently uncertain one [2]. Users may find difficulty expressing their information need into an appropriate request for the retrieval system and they may not have a good idea of what information is available for retrieval. The concept of relevance feedback had arisen from the observation that although searchers have difficulties formulating retrieval queries, they can recognise relevant documents when the documents presented contain useful information. Relevance information can be exploited quantitatively - retrieving more documents similar to the relevant ones - and qualitatively - ranking higher documents that better match relevant ones [2].

The ostensive model of cognition described in [3] relates changes in the knowledge state of a user in response to information encountered during information seeking activities. The core components of the model are shown in Figure 1.

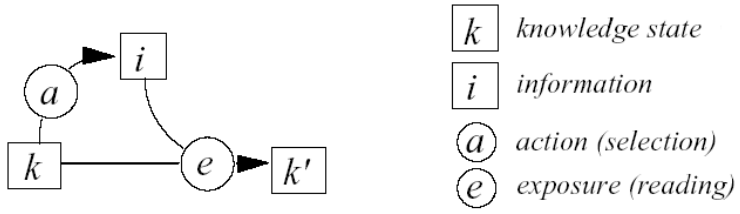

Fig. 1. The updating of a knowledge state through the selection of, and subsequent exposure to information, taken from [3].

According to this model the *knowledge state* k updates to *knowledge state* $k\prime$ subject to *selection action* a and *exposure* e. The update process can continue into an iterative loop with state k updating to state $k\prime$ then to $k\prime\prime$ and so on. Mapping this model to a retrieval system we conclude that the *knowledge states* $k, k\prime$ .. are specific to both users and system (for every *knowledge state* k of the user there is a virtual *knowledge state* k of the system), the *action* a is performed by the retrieval system when retrieving the document i and the *exposure* e is performed by the user. Going further into the approach taken in [3] that discusses

only the user perspective, we consider simultaneously the system perspective. We argue that *action a* - the retrieval of a document by the system - and *exposure e* - the feedback returned by the user - are observable, but the *knowledge states k* are non-observable. The user can see that a particular document has been retrieved but may not understand the system state that triggered the selection of that specific document. Similarly, the system, receiving feedback, is instructed about the relevance of the document without being informed in which way that document is relevant to the user.

In terms of video shot retrieval based on objects, there are many parallels with this model in that users move from one knowledge state to another based on exposure to some video clip or shot. We would argue that it is even more the case in video shot retrieval, based on objects, that users will not understand the system's reasoning as to why one shot may have been retrieved and that the system should be given something more than just yes/no relevance judgments on shots/objects and that relevance feedback should be faceted where possible. In the work we report in this paper we shall show how we achieve just that.

Researchers in the field of text retrieval have experimented with explanations as a technique "to reduce the conceptual gulf between how the system operates and how the user thinks the system operates" [4], [5]. In retrieval systems the explanatory power has been traditionally exploited by two models: the dialogue model [5], and domain knowledge representation [6]. The dialogue model controls what is to be explained and at what stage, whereas the domain knowledge determines the content of the explanation. However both models have drawbacks: domain knowledge representation is hard to achieve on heterogenous data collections [4], and there is strong indication that most users do not follow the search strategies proposed to them by the dialogue model [7].

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. 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 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.

3 Video Shot Retrieval Algorithm

In this section we give an outline of our algorithm for video shot retrieval based on multiple example query objects. 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:

$$S_{object}(i,j) = \alpha S_{colour}(i,j) + \beta S_{shape}(i,j) + \gamma S_{texture}(i,j) \tag{1}$$

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.

For each feature 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:

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

The mean μ and variance Σ are estimated from examples labelled as relevant (positive). We consider the optimal model to be the Gaussian mixture with the minimum number of components that correctly classifies the labelled set of objects. This means that any object labelled as a positive instances is within one standard deviation away form the mean of at least one Gaussian component and that no object labelled as a negative instances is within two standard deviations away form the mean of any Gaussian component. The expectation-maximisation (EM) algorithm [8] is employed to estimate the density probability functions of the Gaussian mixture.

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

$$\begin{aligned}
\overline{v}_{colour} &= ((\mu^{(1)}_{colour}, \Sigma^{(1)}_{colour}),..,(\mu^{(n)}_{colour}, \Sigma^{(n)}_{colour})) \\
\overline{v}_{shape} &= ((\mu^{(1)}_{shape}, \Sigma^{(1)}_{shape}),..,(\mu^{(m)}_{shape}, \Sigma^{(m)}_{shape})) \\
\overline{v}_{texture} &= ((\mu^{(1)}_{texture}, \Sigma^{(1)}_{texture}),..,(\mu^{(p)}_{texture}, \Sigma^{(p)}_{texture}))
\end{aligned} \tag{3}$$

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.

$$query_{(i,j,k)} = ((\mu^{(i)}_{colour}, \Sigma^{(i)}_{colour})(\mu^{(j)}_{shape}, \Sigma^{(j)}_{shape})(\mu^{(k)}_{texture}, \Sigma^{(k)}_{texture})) \tag{4}$$

Each query triplet is a possible search direction and is displayed in the user interface by grouping together the video objects that belong to this triplet (see Section 5). There is a possibility of the number of queries growing exponentially

with the number of features and so we limit the expansion of triplets by introducing a "mixture expansion factor" that constrains the increase in the number of components in the Gaussian mixture. The expansion factor is $1/N$, where N is the number of existing component in the Gaussian mixture. As N increases, the expansion factor becomes smaller therefore inhibiting the addition of new components. 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 categorise user query objects where each query category could represent a set of objects which are similar to each other but dissimilar to other query objects. So, for example, if a user is searching for motor car objects then one category could be red VW Beetle objects and another category could be white jeep objects and the two categories of objects will have different colours (red or white) and shapes (Beetles are more curved in shape than jeeps), though textures may be similar.

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 estimation-maximisation and query construction steps are repeated when new examples are labelled by the user.

4 Feature Descriptors for Objects

The features 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 (segmented object). We realise that image background conveys important information as well, but we do not consider this in this investigation.

4.1 Colour Representation

To represent colour we adopted the MPEG-7 Dominant Colour Descriptor (DCD) [9], which is used by many retrieval systems. The recommended distance to be used with DCD is [10] :

$$D_{DCD}(Q,I) = \left(\sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2\alpha_{1i,2j} p_{1i} p_{2j} \right)^{1/2} \tag{5}$$

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:

$$\alpha_{k,l} = \begin{cases} 1 - \frac{D_{k,l}}{D_{max}}, & D_{k,l} \leq T_d \\ 0, & D_{k,l} > T_d \end{cases} \tag{6}$$

In expression 6 $D_{k,l} = \| c_k - c_l \|$ 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 [11].

4.2 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 [12], defined by Equation 7. This is a simple and robust descriptor that can indicate a degree of shape similarity.

$$\gamma = \frac{P_2}{4\pi A} \tag{7}$$

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

$$P = \sum_{i=1}^{N-1} \parallel x_{i+1} - x_i \parallel + \parallel x_N - x_1 \parallel \tag{8}$$

4.3 Texture Representation

In our system texture is represented with the MPEG-7 Texture Browsing Descriptor [9]. This descriptor is expressed as a set of 24 Gabor wavelets [13] $g_{m,n}(x, y)$ (6 orientations, 4 scales) obtained by appropriate rotations and dilations of the a two dimensional Gabor function:

$$\begin{aligned} &g_{m,n}(x, y) = a^{-m} G(x', y'), \quad a > 1 \\ &x' = a^{-m}(x \cos\theta + y \sin\theta) \\ &y' = a^{-m}(-x \sin\theta + y \cos\theta) \end{aligned} \tag{9}$$

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:

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jW\right] \tag{10}$$

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

$$W_{m,n}(x, y) = \int\int I(x', y') g_{m,n} * (x - x', y - y') dxdy \tag{11}$$

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 [13]:

$$\begin{aligned} \mu_{m,n} &= \int\int |W_{m,n}(x, y)| dxdy \\ \sigma_{m,n} &= \sqrt{\int\int (|W_{m,n}(x, y)| - \mu_{m,n})^2 dxdy} \end{aligned} \tag{12}$$

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 [13] is defined as:

$$\begin{aligned} d(i,j) &= \sum_m \sum_n d_{m,n}(i,j) \\ d_{m,n}(i,j) &= |\frac{\mu_{m,n}^{(i)} - \mu_{m,n}^{(j)}}{\alpha(\mu_{m,n})}| + |\frac{\sigma_{m,n}^{(i)} - \sigma_{m,n}^{(j)}}{\alpha(\sigma_{m,n})}| \end{aligned} \tag{13}$$

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.

5 User Interface and System Interaction

In this section we focus on the front end side of the system we have developed to allow the user to select query objects and to include query branching where the system offers two or more diverging queries for the user to pursue (red VWs and white jeeps in the example earlier).

We start with a description of the design scheme we developed to allow the user to browse and specify a particular object within an image content and use only that object for subsequent querying; then we describe how this scheme has been incorporated into an overall interface in which the interactive search stages (browsing, collecting relevant objects, querying based on the objects and re-querying) are implemented.

5.1 Interacting with Objects

Objects automatically detected by the system should be visible to the user in some way, so that s/he could see what possible further interaction can be done with it.

Figure 2(a) shows a keyframe, in this case showing a white car. An oval button on the right represents the detection of an object within the image with its three low-level features. If there is more than one object detected in the image, there will be a button for each.

In Figure 2(b), the user highlights the objet of interest by selecting the button though in our current implementation there is a maximum of one object per keyframe. In Figure 2(c), after selecting an object the user then can specify which low-level features (colour, shape or texture) of the specified object s/he is interested in. Each of the feature buttons toggles between positive, negative or neutral for each feature of the object. Once feature indications are specified, the user can copy this object (and its specified features) to the query panel as shown in Figure 2(d) where the image contains only the specified object with the background stripped away. The feature specification for this object will be now used for relevance feedback.

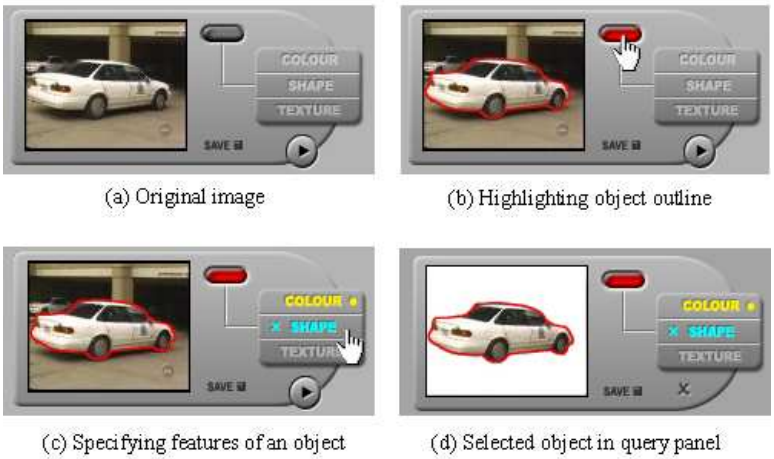

Fig. 2. Object visualisation and interaction

5.2 Relevance Feedback Using Objects

The main features of the interface to our object-based video retrieval system are for the user to:

- Browse initial set of objects;
- Specify particular features of an object to use for relevance feedback;
- Browse a number of user-selected objects and their specified features to adjust, remove, and add to the set of query objects;
- Trigger retrieval based on the specified features of the query objects;
- Browse retrieved objects and use some as additional feedback;
- Save relevant objects in a separate folder

In addition to the above, an important feature of our system is to allow the user to view how his/her relevance feedback and set of query objects is semantically consistent/inconsistent by showing clusters within the set of query objects. If this set of query objects is not visually consistent, using all of this feedback for retrieval will confuse the system and lower the retrieval accuracy. This is similar to adding very visually different image examples in Query-By-Example systems. Although a syntactically legitimate action by the user, this behaviour results in degraded retrieval and thus contributes negatively to the interaction. Thus, if the system can split the relevance feedback and set of query objects into semantically coherent query object groups and present them to the user, s/he can identify this and "branch" the query into two or more and then focus on only one of the groups at a time. This maps back nicely to the ostensive model of retrieval where a user wishes to pursue two or more "lines of enquiry" but can only do one at a time. With our approach, where the system

has partitioned the search into two or more distinct clusters, one can be pursued as the set of query objects while the other cluster(s) is put on hold and returned to at a later stage. A worked example illustrates this.

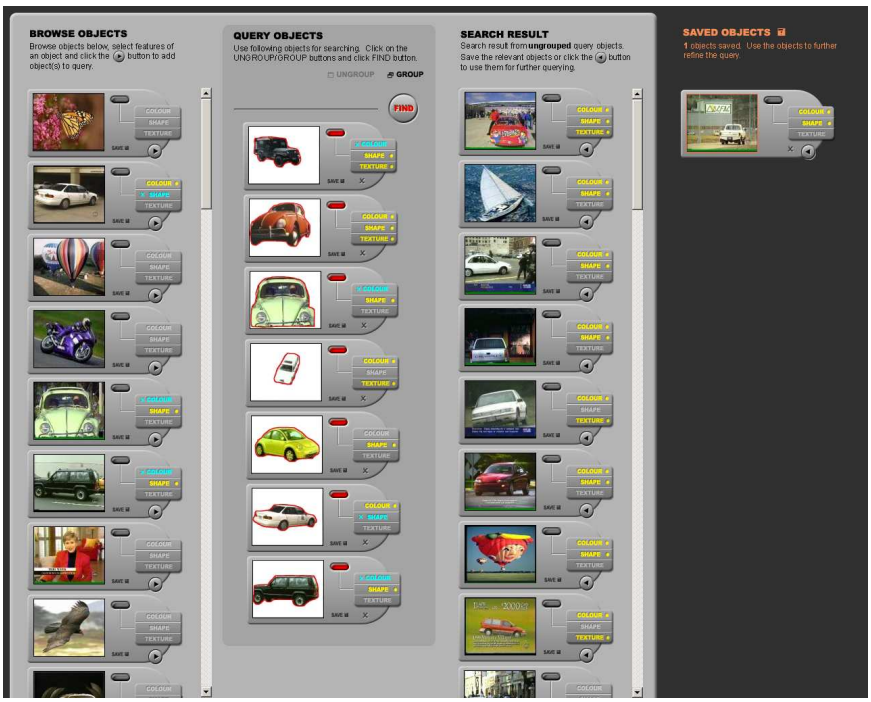

Fig. 3. Overall interface - query panel (2nd column) contains the 7 objects that the user selected during the interaction, and the 3rd column shows the search result based on the all 7 objects.

Figure 3 shows a screen shot from the system in which the interface is divided into 4 columns. The first column presents an initial set of images in the representation format described in Section 5.1. The user browses this set of images, views objects and specifies features, then adds some of the objects to the "QUERY OBJECTS" panel (2nd column). A similar interface facility can be found in numerous experimental image and video retrieval systems in which the user can select example images to be used for subsequent queries as a mechanism for relevance feedback, as in [15], [16], [17], [18], [19], however, unlike these systems, the added examples in our system are objects, not a whole image or an image region. Figure 3 currently shows 7 objects added to the query panel. Clicking on the "FIND" button triggers retrieval based on the 7 objects and the positive, negative or neutral indicators of their features, and the result is presented on the "SEARCH RESULT" panel (3rd column). If a relevant object is found in the search result, the user can save it to the "SAVED OBJECTS" panel (4th column).

The user can also add more objects to the query panel from the search result, or from the saved object panel. As the user browses, searches and saves more and more relevant objects, s/he can collect more relevant objects into the query panel.

At the top of query panel (2nd column), the user can click on the "GROUP" button to view how the system can internally split objects in the query panel. This split of query objects is displayed in Figure 4.

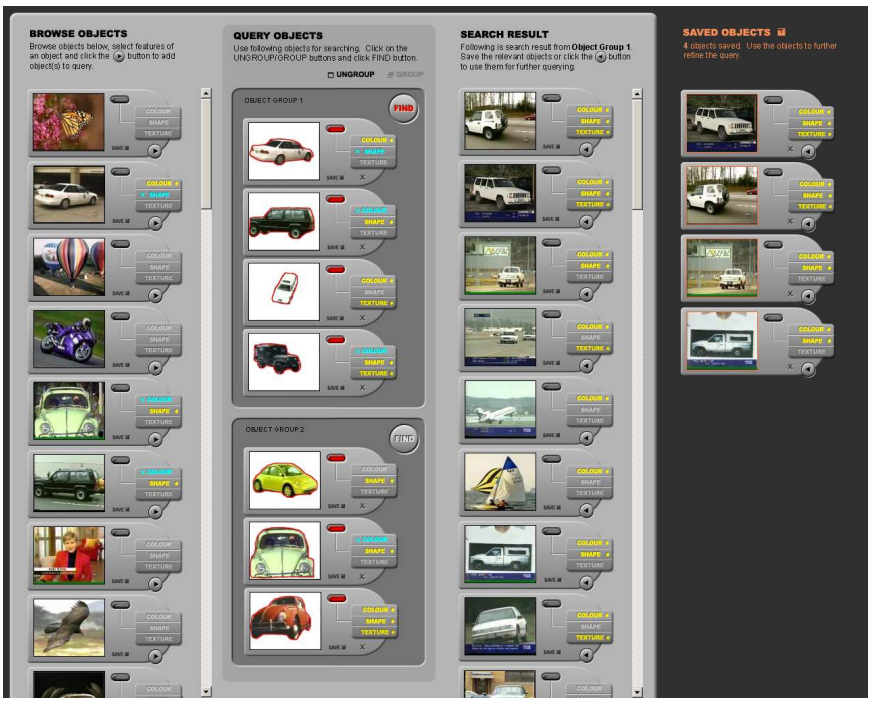

Fig. 4. Query objects are split into 2 groups as the result of the system's internal clustering. The user can now pursue searching either or both of the query groups.

In Figure 4, the 7 objects the user added to the query have been split into 2 groups according to the system's clustering algorithm. The user can now see how s/he has been adding objects of two different types: in the 1st group (top 4 objects in the 2nd column), the object characteristics indicate white colour, more square shaped vehicles such as a white jeep and in the 2nd group (bottom 3 objects in the 2nd column), the object characteristics indicate red, round shaped vehicles such as a VW Beetle, quite different from the one formulated in the first group. As this split among the added objects is now revealed to the user, s/he can decide to focus on searching for only one type of object (either 1st or 2nd group) to find more objects that are like only either of the groups. In Figure

4 the user searched on the 1st group by clicking on the "FIND" button of that group, and the top few search results show white cars with squared edges. The fact that the search result is from the 1st group of objects is indicated by the "FIND" button in that group and mentioned at the top of the search result. At this stage, when the user adds another object into the query panel it will be automatically inserted into the 1st group if the added object is deemed to be in the same cluster as the 1st group; or inserted into the 2nd group if deemed more similar to 2nd cluster; or as a separate, 3rd group in the case that it is far from the feature space of the either groups.

In this way, the user can see semantic clustering of query objects as s/he adds and specifies the features of objects, and can conduct a more multi-threaded search by pursuing one of the clusters of query objects at a time. Inconsistent relevance feedback is still a legitimate action by the user but our system is adaptive in that it suggests a better way of searching by automatically splitting the relevance feedback history into semantically coherent clusters so that the user can continue with a more consistent subset of his/her own feedback objects and can search query object clusters, one at a time. As mentioned earlier, this maps neatly to one aspect of the ostensive model for retrieval where a user is confronted with two distinct threads to their search which they wish to pursue in sequence, both falling under the one information need. By automating the detection of these threading or branch-off points and maintaining both such threads as separate, live searches, the user is encouraged to follow his/her own instincts if these match the threads suggested by the system.

6 Conclusions

In this paper we introduced an object-based video search system that features interactive query formulation using the colour, shape and texture of an object, and through iteration of query/browsing, the system incrementally improves modelling of video objects. The actual segmentation of objects from keyframes in our system was semi-automatic and supervised in order to provide accurate object sets and to better illustrate our retrieval approaches in which the matching among objects (i.e. relating all similar objects in the database) can be helped using the user's query formulation history as feedback.

The status of our work is that we have build the retrieval system described in this paper and we have a collection of video with 650 semi-automatically segmented objects, we have completed the user interface as described here, we have completed some initial user testing and we are starting a more comprehensive interactive user testing and evaluation.

We are also working on several improvements including making object segmentation from each keyframe fully-automatic. Segmenting more than one object from each keyframe is also part of our future work; 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.

7 Acknowledgments

The support of the Enterprise Ireland Informatics Initiative is gratefully acknowledged. Part of this work was supported by Science Foundation Ireland under grant 03/IN.3/I361

References

1. A.F. Smeaton, P. Over and W. Kraaij: TRECVID: Evaluating the Effectiveness of Information Retrieval Tasks on Digital Video. in: *Proceedings of the 12th ACM International Conference on Multimedia 2004*, pp. 652-655, New York, NY, 15-16 October 2004.
2. I. Ruthven and M. Lalmas: A Survey on the Use of Relevance Feedback for Information Access Systems. *Knowledge Engineering Review*, Vol. 18, pp. 95-145, 2003.
3. I. Campbell and C.J. van Rijsbergen: The Ostensive Model of Developing Information Needs. In: *Proceedings of the 2nd International Conference on Conceptions of Library and Information Science*, CoLIS 2, Copenhagen, Denmark, 1996.
4. I. Ruthven: On the Use of Explanations as a Mediating Device for Relevance Feedback. in: *Proceedings of the 6th European Conference on Digital Libraries*, ECDL 2002, Lecture Notes in Computer Science, Rome, 2002.
5. N.J. Belkin: On the Nature and Function of Explanation in Intelligent Information Retrieval. in: *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Grenoble, 1988.
6. A. Cawsey: Explanation and Interaction: the Computer Generation of Explanatory Dialogues. MIT Press (The ACL-MIT Press Series in Natural Language Processing), 1992.
7. S. Dennis, R. McArthur and P. Bruza: Searching the WWW Made Easy ? The Cognitive Load imposed by Query Refinement Mechanisms. in: it Proceedings of the Third Australian Document Computing Symposium, 1998.
8. T. K. Moon: The Expectation-Maximisation Algoritm. *IEEE Signal Processing Magazine*, pp. 47-60, November 1996.
9. P. Salambier and J.R. Smith. MPEG-7 Multimedia Descriptions Schemes. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, pp. 748-759, June 2001.
10. B. Manjunath, P. Salambier and T. Sikora: Introduction to MPEG-7: Multimedia Content Description Interface. Wiley, New York, USA, 2002.
11. A. Kushki, P. Androutsos, K.N. Plataniotis and A.N. Venetsanopoulos: Query Feedback for Interactive Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, Issue 5, pp. 644-655 , May 2004.
12. S. Theodoridis and K. Koutroumbas. Pattern Recognition. Academic Press, 1999.
13. B.S. Manjunath and W.Y. Ma: Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, Issues 8, pp. 837-842, August 1996.
14. I. Ruthven, M. Lalmas and K. van Rijsbergen: Ranking Expansion Terms Using Partial and Ostensive Evidence. in: *Proceedings of the 4th International Conference on Conceptions of Library and Information Science*, CoLIS 4, Seattle, 2002.

15. Y. Lu, Ch. Hu, X. Zhu, H.J. Zhang and Q. Yang: A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems, in: *Proceedings of ACM Multimedia*, pp. 31-37, 2000.
16. D. Heesch and S. Rüger: Three Interfaces for Content-Based Access to Image Collections. in: *Proceedings of the Interantional Conference on Image and Video Retrieval*, CIVR 2004, Dublin, pp. 491-499, 2004.
17. M. Worring, G.P. Nguyen, L. Hollink, J. van Germert, D.C. Koelma: Interactive Search Using Indexing, Filtering, Browsing, and Ranking. in: *Proceedings of the TRECVID Workshop*, Gaithersburg, Maryland, 15-16 November, 2004.
18. E. Cooke, P. Ferguson, G. Gaughan, C. Gurrin, G. Jones, H. Le Borgne, H. Lee, S. Marlow, K. McDonald, M. McHugh, N. Murphy, N. O'Connor, N. O'Hare, S. Rothwell, A.F. Smeaton and P. Wilkins: TRECVID 2004 Experiments in Dublin City University. in: *Proceedings of the TRECVID Workshop*, Gaithersburg, Maryland, 15-16 November, 2004.
19. C. Carson and H. Greenspan: Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, Issue 8, 2002.