VIDEO RETRIEVAL USING DIALOGUE, KEYFRAME SIMILARITY AND VIDEO OBJECTS

Paul Browne[1] *and Alan F. Smeaton*[1,2]

Centre for Digital Video Processing[1] and Adaptive Information Cluster [2]
Dublin City University,
Glasnevin, Dublin 9, Ireland.

ABSTRACT

There are several different approaches to video retrieval which vary in sophistication, and in the level of their deployment. Some are well-known, others are not yet within our reach for any kind of large volumes of video. In particular, object-based video retrieval, where an object from within a video is used for retrieval, is often particularly desirable from a searcher's perspective. In this paper we introduce FÍSCHLÁR-SIMPSONS, a system providing retrieval from an archive of video using any combination of text searching, keyframe image matching, shot-level browsing, as well as object-based retrieval. The system is driven by user feedback and interaction rather than having the conventional search/browse/search metaphor and the purpose of the system is to explore how users can use detected objects in a shot as part of a retrieval task.

1. OVERVIEW OF VIDEO RETRIEVAL

Current approaches to content-based retrieval of video can be broadly divided into those which support text matching against dialogue, image matching against keyframes, or which detect concept features in order to filter shots. This observation is based on the annual TRECVid exercise [4] which, for each of the last four years, has benchmarked the effectiveness of video retrieval tasks on a shared video corpus using a metrics-based approach. During 2004, over 30 research groups took part in this retrieval exercise so the different approaches taken in TRECVid represent a broad sweep of research in the field.

The first approach to video retrieval, text matching, matches a user's text query against text obtained from video such as from closed captions or from recognized speech, or from video OCR. Conventional text-based information retrieval approaches can be used and term weighting approaches such as BM25 or IDF have been applied with success [4]. Text-based video retrieval works well where the focus of the search is the item under discussion in the video but when the focus of the search is the visual content itself then the second approach to video retrieval, image matching, is more suitable.

The simplest kind of image matching uses low-level image features such as global or local colour, texture, edges or shapes [5]. In video retrieval, a single keyframe is usually chosen as the representative image for a shot and often it is the middle frame from a shot that is chosen. Using this approach it is not surprising that matching a user's query image(s) against keyframes using low-level features can be successful only if the user's information need can be captured in a still image and the keyframe is genuinely representative of the shot. Searches for shots of a "rocket launch" for example, with a query image consisting of a blaze of flames coming from a rocket against a blue sky background will be visually similar to keyframes taken from rocket launch footage, so there are also cases when this approach to retrieval works well.

The automatic detection of semantic concepts in video represents a third approach to video retrieval. This involves manually annotating a large corpus of shots as either having or not having feature "X" and then using a machine learning algorithm to build an automatic classifier for this. Examples of such features used in TRECVid [4] are "indoor", "outdoor", "buildings", etc. Building concept feature detectors is a difficult task because the definition of a set of concepts broad enough to cover a large range of information needs yet discriminative enough to be useful in user searching is an open question and the task of labeling enough positive and negative examples of a concept and then learning a classifier is something that requires much effort. Nevertheless, if a concept feature detector is available and has been applied to a corpus of video with a high level of accuracy then it can certainly be useful in a user's search. The problem is that there are only a small number of such classifiers built and many have a level of accuracy which is less than that required for reliable video retrieval.

With a range of search modalities available for video retrieval, and each working best only on certain types of

The support of the Informatics Directorate of Enterprise Ireland is gratefully acknowledged. This work was partially supported by Science Foundation Ireland under grant No. 03/IN.3/I361.

0-7803-9134-9/05/$20.00 ©2005 IEEE

search, clearly the best overall approach is to make as many of these as possible available to a user and let the user use some or all of them, in some weighted combination. A recent approach by Yan et al. [6] has been to assign a users search to a pre-defined type and for each the optimal combination of retrieval tools such as those outlined above will have been learned. This appears to be accepted as the best overall approach to video shot retrieval.

Retrieval based on the presence, or absence, of given objects in the video has not received much attention to date. Sometimes we want to retrieve video shots based on an identifiable object in the video such as a car, a dog, the Eiffel Tower or a motorbike. In such cases we could hope that the dialogue will mention what is on-screen but we cannot rely on this. We could hope that reliably detected and useful features will help narrow down the corpus so we can browse the remainder but we cannot rely on this nor can we rely on matching an entire example image containing our desired object against a keyframe based on colour, texture or edges. A quantum leap in video retrieval would be made if a searcher could specify one or more visual objects, segmented from their backgrounds, and use those as part of the query. In the next section we shall see how such object detection and segmentation can be done.

2. OBJECT DETECTION, SEGMENTATION AND TRACKING IN "THE SIMPSONS"

Despite much focus and attention, fully automatic object detection, segmentation and tracking across video is not yet achievable for natural video and available systems require user involvement and are thus semi-automatic at best. What this means in practice is that such systems are assisted by a human who draws a rough boundary around an object of interest and the system will detect and segment this object, separating it from its background, and track it temporally through the shot [1].

Object detection from natural video needs to be invariant to scale and to rotation. Part of the object may be occluded either by another object or by the object itself, and variations of the same object type may generate very different object images, such as a car object for example. Detection will also have to deal with noise such as edges from other objects in the image and/or errors in the edge detection process itself. Finally, images and video frames are 2D representations of a 3D world without depth information and for a symmetrical object like a vase it doesn't matter what angle we look at it from as the shape will always be the same, but most objects are not like this which means a car object, for example, can have very many different shapes depending on the angle of view.

Our approach to segmentation is to use object templates where a number of templates of the object to be detected are

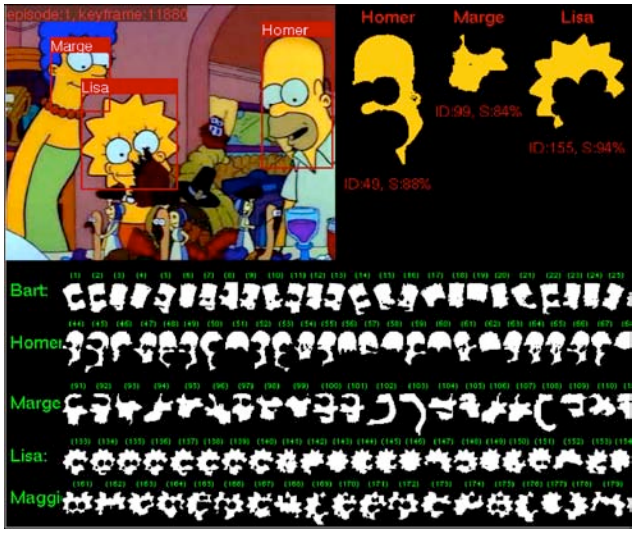

Fig. 1. Screenshot to illustrate template matching

pre-defined and shape matching is used during the detection to compare each image against the available templates [1]. This performs well on animated content where there are a finite and limited number of views of objects, objects do not turn and the camera usually has fixed positions.

We automatically detected the presence of the major characters appearing on-screen in a collection of Simpsons episodes by extracting all yellow coloured objects from shot keyframes (all Simpsons characters are yellow) and matching these extracted objects against a number of previously constructed templates for each character. In the detection process each template was compared against shape boundaries in the comparison image using a shape matching algorithm which generates a number of comparison scores and any scores above a pre-defined threshold were regarded as a positive match. Colour plays an important part in the shape matching process, as positive matches require yellow areas to match the templates. All other colours are ignored and this reduces false detection.

Figure 1 is a screenshot taken from an interface to the template-based object detection application we developed [1]. At the bottom of the screen we can see 5 members of the Simpsons family and their representative templates. At the top left of the screen we can see the keyframe image that is being compared while the top right shows positive template matches for Homer, Lisa and Marge. The overall accuracy of our template-matching detection needs to be high in order for subsequent searching to be reliable. To measure this we used a development corpus of 12 episodes (4.8 hours, 6,525 shots) and manually tagged each shot which had any of our target characters (objects). Table 1 gives a list of the 10 characters who were detected automatically using this approach, their occurrence in the development corpus, and

Table 1. Automatic object detection in the Simpsons corpus of 6,525 shots

Character	Occurrences	Recall	Precision
Homer	2,066 (32%)	0.36	0.98
Marge	1,343 (21%)	0.32	0.87
Bart	1,361 (21%)	0.49	0.98
Lisa	647 (10%)	0.57	0.93
Maggie	321 (5%)	0.31	0.88
Mr. Burns	269 (4%)	0.47	1.0
Mr. Smithers	175 (2.6%)	0.46	0.98
Abe Simpson	83 (1.3%)	0.27	0.91
Principal Skinner	246 (4%)	0.8	1.0
Moe	43 (0.7%)	0.61	0.78
Avg. (weighted)		**0.421**	**0.947**

the accuracy of detection measured against our manually tagged ground truth using precision and recall. The results of our tests show that precision at 0.947 is high for almost all characters, but recall at 0.421 is low, indicating we are missing many character occurrences. These could be due to the character not being present or perhaps occluded in the keyframe but present in some other part of the shot. While at first glance this may seem unacceptable, we should note that the presence of the minor characters in the series (i.e. not Bart, Homer or Marge) tends to occur in clusters or scenes so even if a given character occurrence is missed, it is likely that occurrences in neighbouring shots will be detected and picked up by the user when browsing the the video.

3. THE FÍSCHLÁR-SIMPSONS VIDEO RETRIEVAL SYSTEM

3.1. Video Retrieval

Because it is visual in nature, video information is intrinsically browsable and invites a browsing-style interaction from users seeking information. A typical search interaction involves a search generating a ranked list of documents, which the user previews and chooses some document to read and this continues until the user decides to re-formulate the query and the process re-iterates. In our work we use an interaction metaphor in which a user's initial query generates a ranked list of shots, the user browses these, examines one in more detail and judges the shot as one of the set of relevant (R_S) or the set of non-relevant ($\overline{R_S}$) shots or judges some object in the shot as one of the set of relevant (R_O) or the set of non-relevant ($\overline{R_O}$) objects. Based on this judgment the shots are then dynamically re-ranked and the user is given a re-ranking to browse and choose another shot and make a second shot or object relevance judgment. All shot or object relevance judgments, plus the initial query, are used to re-rank shots and the process of dynamically re-ranking after every judgment continues until the user is satisfied.

3.2. Ostensive Relevance Feedback

In the interaction scenario outlined above, the system uses a number of shot and/or object relevance judgments to generate a shot ranking. In our work we use *ostensive relevance feedback* [3], where the importance of a shot or object relevance judgment is a function of the "age" of the judgment with more recent judgments having more importance and "age" being the relative order of shot judgments as received by the system, measured in terms of the user's current search session. The premise behind this is that it can capture the shifting nature of a user's information need, especially in a highly browsing-intensive application such as video searching. A number of decay functions for weighting older shots by age have been used in our work including linear and logarithmic and for the purposes of the work described here we use a linear decay. In the case where a user has viewed a number of shots and found k relevant shots, l non-relevant shots, m relevant objects and n non-relevant objects, an unseen shot C_u is ranked by the function Sc defined as:

$$Sc(C_u) = S(C_u, Q) \; + \sum_{i=1}^{k} S(C_u, X_i^{rel})\, OD_i - \sum_{j=1}^{l} S(C_u, X_j^{non})\, OD_j + \sum_{i=1}^{m} S(C_u, Y_i^{rel})\, OD_i - \sum_{j=1}^{n} S(C_u, Y_j^{non})\, OD_j$$

where $S(S_1, S_2)$ is a function which measures the similarity between two shots S_1 and S_2, Q is the original query (interpreted as shot) and $X_i^{rel} \in R_S$, $X_j^{non} \in \overline{R_S}$, $Y_i^{rel} \in R_O$ and $Y_j^{non} \in \overline{R_O}$. The weights OD_i and OD_j are the ostensive decays of the previously judged shots and objects and are scaled linearly between 0 and 1.

3.3. Video Shot Similarity

Computing a similarity $S(S_1, S_2)$ between two shots is done using text-text, image-image and object-object metrics. For text, we use common stopwords weighted by their inverse document frequency, the number of shots in the corpus (N) divided by the number of those for which the stopword occurs (n_i). This is defined as $S_{text}(S_1, S_2) = \sum W_1 \times W_2$ where $W_x = log_2(N/n_x)$.

For image similarity we use the sum of the absolute difference in feature values for a range of low-level image features including average colour over 9 frame regions, median colour over each of 9 frame regions, an 18-bin hue histogram for 4 frame regions and a 16-bin edge histogram

for 4 frame regions. These absolute differences in feature values are divided by their absolute sums and can be considered as a relative Manhattan distance.

$$S_{img}(S_1, S_2) = \sum_{i-1}^{k} [\frac{|x_{i,j} - x_{k,j}|}{|x_{i,j} + x_{k,j}|}]$$

The final component of our shot matching, using objects, uses a set of pre-defined static weights as additions or subtractions from the shot-shot similarity, depending on the presence of common objects in the set of relevant (R_O) or non-relevant ($\overline{R_O}$) objects respectively. Finally, the three elements of our shot similarity, $S_{text}()$, S_{img}, and object similarity, are linearly combined with the overall shot similarity formula incorporating the ostensive decay weights.

3.4. The FÍSCHLÁR-SIMPSONS Video Retrieval System

The TV show "The Simpsons" now has over 340 episodes (136 hours of 5.6 days of content) and for our video corpus we used 52 of these (20.8 hours or 20,529 shots) taken from seasons 2 and 3, and themed DVDs covering seasons 4 to 12. These were transcoded into MPEG-1 for image analysis, storage and playback. For closed captions the DVD subtitle text was stored as a series of images and therefore optical character recognition (OCR) was required to convert these into machine-readable text.

In preparing our video content we first applied a shot boundary detection algorithm to this corpus [2] and for each shot we identified a keyframe. We then built a video retrieval system to support the search and shot/object feedback interaction described earlier. Once a user's search is underway it is important to be able to view the context of a ranked shot. In many cases a query topic might have relevant shots near the same location of a ranked shot, therefore it is important to be able to view the shot context (shots that came before and after a ranked result). The *Shot Context screen* (Figure 2) shows the context (preceding 5 shots and following 5 shots) of a given shot shown in the centre of the screen and shows how detected objects are presented to the user.

4. SUMMARY

The key interest in the system developed here is to explore how useful object selection can be as part of video shot retrieval which we do by combing object-retrieval with text-based and image-based shot retrieval under a single ostensive relevance feedback framework. Our future work will involve evaluating this system with real users in a search task environment to gauge exactly what contribution object-based searching can make.

Fig. 2. Shot context screen from FÍSCHLÁR-SIMPSONS

5. REFERENCES

[1] T. Adamek and N. O'Connor. "Efficient Contour-based Shape Representation and Matching," *MIR 2003 - 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, Berkeley, Ca, 7 Nov. 2003.

[2] P. Browne, A.F. Smeaton, N. Murphy, N. O'Connor, S. Marlow and C. Berrut. "Evaluating and Combining Digital Video Shot Boundary Detection Algorithms," *Proc. of the Irish Machine Vision and Image Processing Conference*, Belfast, N. Ireland, 31 Aug.-2 Sept. 2000.

[3] I. Campbell, "Interactive evaluation of the Ostensive Model using a new test collection of images with multiple relevance assessments," *Journal of Information Retrieval*, **1**, pages 85-112, 2000.

[4] A.F. Smeaton, W. Kraaij, and P. Over. "The TREC Video Retrieval Evaluation: A Case Study and Status Report," *RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*, Avignon, France, 26-28 Apr. 2004.

[5] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain. "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. Pattern Anal. Mach. Intell.* 2000, **22(12)**, pp. 1349-1380.

[6] R. Yan, J. Yang. and A.G. Hauptmann. "Learning Query-Class Dependent Weights in Automatic Video Retrieval," *Proc. of the ACM Multimedia Conference*, New York, Oct. 2004, pp.548-555, ACM Press.