Three Interfaces for Content-Based Access to Image Collections

Daniel Heesch and Stefan Rüger

Department of Computing, Imperial College
180 Queen's Gate, London SW7 2BZ, England
{daniel.heesch,s.rueger}@imperial.ac.uk

Abstract. This paper describes interfaces for a suite of three recently developed techniques to facilitate content-based access to large image and video repositories. Two of these techniques involve content-based retrieval while the third technique is centered around a new browsing structure and forms a useful complement to the traditional query-by-example paradigm. Each technique is associated with its own user interface and allows for a different set of user interactions. The user can move between interfaces whilst executing a particular search and thus may combine the particular strengths of the different techniques. We illustrate each of the techniques using topics from the TRECVID 2003 contest.

1 Introduction

Being able to endow systems with the capacity to exhibit intelligent, human-like behaviour has been an early hope of computer scientists. The past fifty years have seen the gradual erosion of this hope and the consensus seems reached that the early optimism of this research program was ill-founded. The general vision problem, that is the problem of being able to describe the content of a visual scene, is among those problems that have as yet been left untouched by the otherwise relentless progress in computer science. To solve it, we will have to come up with answers to deep and fundamental questions about representation and computation that lie at the very core of human intelligence. This is what renders the problem of content-based image retrieval (CBIR) very exciting and challenging at the same time. The increasing interest in human-computer interaction is testimony of a growing awareness that humans are currently still the most intelligent part of the system and that a tighter integration between humans and machines can lead to results that would otherwise remain unattainable [8]. Unlike in typical computer vision applications, content-based image retrieval (CBIR) systems have an end user seeking information, and thus a dialogue between user and machine seems more adequate from the outset. The presence of a user adds to the problem of image understanding the problem of user understanding, a problem that can evidently only be resolved by incorporating the user in the retrieval process.

P. Enser et al. (Eds.): CIVR 2004, LNCS 3115, pp. 491–499, 2004.

We introduce two techniques for content-based image retrieval, and one technique for content-based image browsing. The first technique uses relevance feedback applied to the retrieval results to update weights of the similarity function. The technique comes with an interface that allows users to give continuous feedback. With the second technique we introduce a novel idea that elegantly bypasses the problem of initial feature weighting by gathering the top-ranked images from a multitude of retrieval processes, each carried out with a different weight set. The resulting set of images, which we call the NN^k of the query (NN for nearest neighbour, and k for the number of features), can each be associated with a weight set that, when chosen, will retrieve similar images. It is thus a two-step process that engages the user in relevance feedback after the first step. The third technique also relies on the NN^k idea, but instead of determining the NN^k for a query at run-time, the NN^k of each image from within a collection are determined beforehand and internal links established between each image and its NN^k. The resulting network provides our basis for content-based image browsing. Details of and evaluative studies on each of the three techniques have been presented elsewhere (e.g. [5], [7], [4]). This paper will place greater emphasis on interface design. We illustrate the functionality of the system by looking in detail how a real search task could be executed using the test collection of TRECVID 2003 and particular search topics thereof. The key contribution of this paper is the development of an integrated interface that combines the strengths of three recently developed techniques for CBIR.

The paper is structured as follows. In Section 2, we briefly discuss work that is related to ours. Section 3 establishes methodological commonalities of the various techniques presented here and includes a brief description of the collection used, the image representations and the method of similarity computation. Section 4-6 introduces the three techniques along with the associated visualizations and user interactions. We summarize and conclude our paper in Section 7.

2 Related Work

Relevance feedback as a particular form of human-machine interaction has become a core paradigm in information retrieval and in particular so in CBIR, where it has been shown to substantially improve retrieval performance (e.g. [9], [10]). Few systems address the problem of how to intelligently weigh features prior to the first retrieval, although it is clear that the efficacy of relevance feedback hinges on a satisfactory first retrieval result as little can be learnt from negative examples alone [5]. Aggarwal et al. [1] have presented an interesting two-step approach to feature weighting. The first step involves modifying the query representation by moving it along each of the feature dimensions, and to regenerate from the thus altered representations a set of query images on which feedback is then given. The second step is the retrieval step itself using the newly learnt weights. The technique appears to improve performance but places constraints on the set of features that can be used. It is methodologically different from but in spirit very similar to our idea of NN^k search.

Visualization of search results has initially taken the form of a 2-d grid layout (e.g. [3]). More recent visualizations aim to preserve the distances of the retrieved images to the query ([5], [13]) and the distances between returned images as in [12]. One of the problems of plane-filling visualizations is the potential overlap between images which we avoid in one of the search result visualizations and minimize in the other.

The importance of browsing as a method of accessing image collections has increasingly been recognized in recent years (e.g. [2], [11]). In the ostensive browsing model developed in [2], the user browses along a dynamically generated tree. The set of possible branches a user may take from a given image depends no less on that image than on the history of past images. The tree is generated dynamically and the method designed to deal with changing information needs. In the model presented by Santini et al. [11], the user can effect a transformation of the image space upon relocating images on the screen. The user can explore the vicinity of a particular region but the scope for fast navigation through the image space is limited. The browsing structure on which our third technique relies on seeks to allow for both the exploration of an image's surrounding as well as efficient browsing, thus allowing for target as well as undirected search.

3 Collection, Features, and Similarity Computation

We illustrate the visualizations using the test collection of TRECVID 2003, a collection that contains more than $32,000$ key frames from news video sequences. We implemented eleven low-level texture and colour features and made use of the text from the speech recognition transcripts supplied by Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur. For details of these low-level features see [4].

The overall similarity between two images X and Y is given by the weighted sum of the feature-specific similarities, i.e. $S(X,Y) = \sum_i w_i s_i(X,Y)$ where the weights are constrained to sum to one with $w_i \in [0,1]$, and s_i are feature-specific similarity functions (the l_1 norm for all features). Because the overall similarity is computed as the weighted sum of these feature-specific similarities, we normalize the feature-specific similarities before aggregation such that their medians lie around 1.

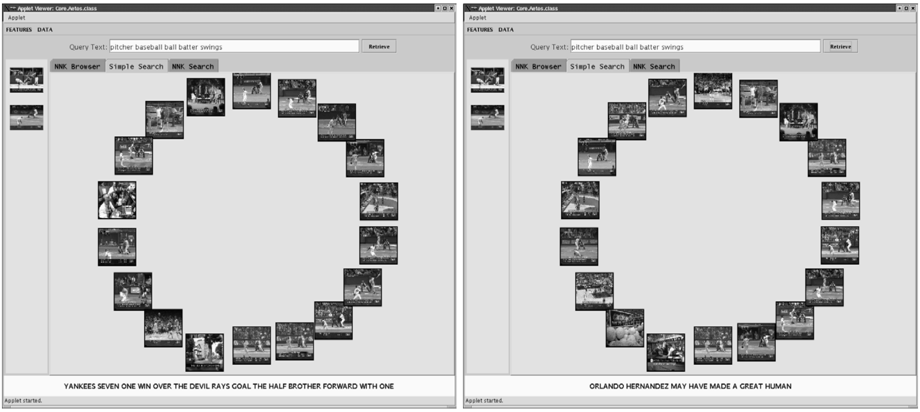
4 Search with Relevance Feedback

4.1 Relevance Feedback Technique

Unlike most systems employing relevance feedback, our technique allows continuous feedback to be given. Given a set of system-computed similarities, the user provides a set of new similarites along a continuous range by relocating images on the screen. We then compute a new set of weights by minimizing the sum of squared errors between the two sets of similarities as described more fully in [5]. The system uses the updated weights for the next retrieval.

4.2 Visualization of Search Results and Relevance Feedback Interaction

Search results are displayed in the form of a spiral with an image's distance from the center being proportional to the distances computed by the system. A similar technique has subsequently been used in [13] where the spiral is constrained to be Archimedean, and the distance of an image to the query is indicated by the distance of the image to the center of the screen along the arc of the spiral. Relevance feedback is given on the displayed images by moving them closer towards or further away from the center. Figure 1 shows the retrieval results with an initially equal weight set and the result following relevance feedback on positive examples (pitcher from behind throwing a ball towards the batter).



Query Text: pitcher baseball ball batter swings

YANKEES SEVEN ONE WIN OVER THE DEVIL RAYS GOAL THE HALF BROTHER FORWARD WITH ONE

Query Text: pitcher baseball ball batter swings

ORLANDO HERNANDEZ MAY HAVE MADE A GREAT HUMAN

Fig. 1. Result before and after relevance feedback using the Baseball topic

Because the initial result set is already quite satisfactory, extensive feedback can be given, leading to a further increase in the number of relevant images.The technique is useful when the search task is relatively easy or once the weights have been brought into the vicinity of the weight optimum using, for example, the NN^k technique to be described in the next section.

5 NN^k Search

5.1 A Two-Step Technique for Feature Weighting

The NN^k idea has first been introduced and explained in greater detail in [6]. We here only give a brief summary of it. Instead of determining similar images to a given query using only one fixed weight set, we determine the *top-ranked* image (the nearest neighbour = NN) for *all possible* weight sets (given k features each associated with a weight, this requires a scan of a k dimensional vector space which can be done very efficiently using a recursive scan of an integer lattice imposed on the weight space). We call this set of top-ranked images the NN^k of a query. The idea is that by not restricting ourselves to one particular weight set, we are able to capture and expose more of the different meanings an image may have. For each nearest neighbour, we record the proportion of the weight space for which it was ranked top. This provides us with a new measure of similarity which will be used in this and the subsequent visualization. We also record the average of all the weights for which a particular NN^k was ranked top.

Determining the NN^k of a query image forms the first step of the technique. The user then gives positive relevance feedback on the set of NN^k. The second step involves retrieving with the weight sets associated with the selected NN^k (the average weight sets mentioned above). The idea is that those weight sets will be likely to cause other similar images to surface. If more than one image has been selected, the ranked lists obtained for the different weights sets are merged. We compared performance of the two-step NN^k search with our own relevance feedback technique [5] and an alternative weight update method by Rui [10] using a small subset of the Corel Gallery 380,000 collection. The results (Figure 2) suggest that the NN^k technique does indeed hold some promise as a way of inferring feature weights at run-time.

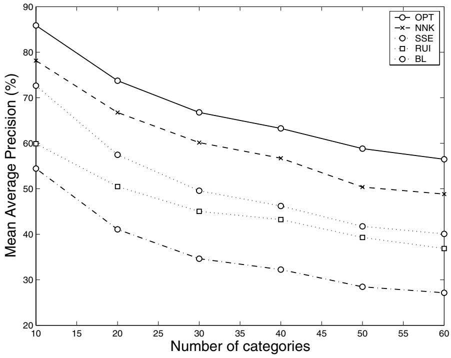


Fig. 2. Mean average precision (MAP) plotted against number of categories for five different retrieval strategies each after one iteration of relevance feedback. Category size is set to 5 and thus the difficulty of the retrieval task increases with the number of categories. Our two-step technique consistently outperforms our own regression method (SSE)[5] as well as Rui's method (RUI) [10]. BL is a baseline obtained by using equal weights for all features. OPT is the optimum performance obtained by using the best weight set for each query

5.2 Visualization and Interaction

For this visualization, we have taken particular care to avoid overlap between images. The results of the two steps are visualized in a similar fashion although the ways the displayed images are obtained differ significantly as we described above. The first step determines the set of NN^k, each with an associated weight set and a similarity value that is proportional to the number of weight sets for which it was ranked top. The layout of images is achieved as follows: images are fitted on the screen in order of decreasing similarity to the query. The first image is displayed in the center, each subsequent image is placed at the first available position that does not produce an overlap. This position is determined by moving outwards from the center such that we trace out an Archimedean spiral. The first position thus found may not be optimal in the sense that the image may be moved still closer towards the center. To achieve a more compact layout, each image is therefore shifted from this initial position towards the center whilst no overlap occurs. Since the number of NN^k can vary considerably with the number of features used and the kind of query image, the size of the images cannot be fixed *a priori*. To ensure that all images fit on the screen, we start with an initially large image diameter and repeat the above procedure with a progressively smaller image diameter until we achieve a complete fit. This adjustment is made whenever the user resizes the window typically resulting in a different configuration of images. This display focusses the user's attention on those images that are more likely to be relevant, with those images that are ranked top for only a small proportion of weight combinations displayed not only in the periphery but also at a correspondingly smaller size. To view peripheral images more clearly, the user can drag any image closer towards the center, where it will be displayed at the same scale as the image the mouse arrow currently points at. The user may now inspect the set of NN^k and, using a pull-down menu, select the most relevant images either to expand the query, or to retrieve with the weight set associated with the selected images.

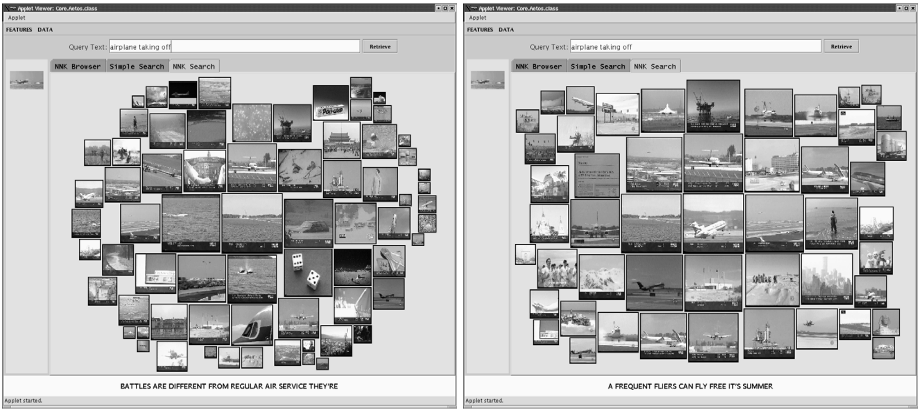
We illustrate the technique using topic 04 from TRECVID 2003 which asks for images depicting planes taking off. Query images are shown on the query canvas. The left picture in Figure 3 shows the NN^k for the query images with 7 planes among them, 2 of which appearing to take off. The user selects the image to the very left, shown in the center of the right screenshot. The number of planes has doubled with 5 out of 14 appearing to take off.

6 NN^k Networks

6.1 Using NN^k for Browsing

It is a natural extension of the NN^k search technique to allow the user to repeat the first step of the NN^k search and determine the NN^k for any of the displayed images (instead of asking for the ranked list produced when using the newly found weight set). This leads to the idea of an NN^k network where the vertices represent individual images and arcs are established between two vertices if one is the NN^k of the other, that is if there is at least one weight set for which one image is the nearest neighbour of the other. Again, we record for each nearest neighbour, the proportion of the weight space for which it was top-ranked.

The advantages of the proposed structure are threefold: first, by looking at a multitude of feature combinations, the network helps expose the semantic richness of images. It is left to any particular user to decide which of the possible interpretations is the most appropriate one by following the corresponding



Query Text: airplane taking off

BATTLES ARE DIFFERENT FROM REGULAR AIR SERVICE THEY'RE

Applet started.

Query Text: airplane taking off

A FREQUENT FLIERS CAN FLY FREE IT'S SUMMER

Applet started.

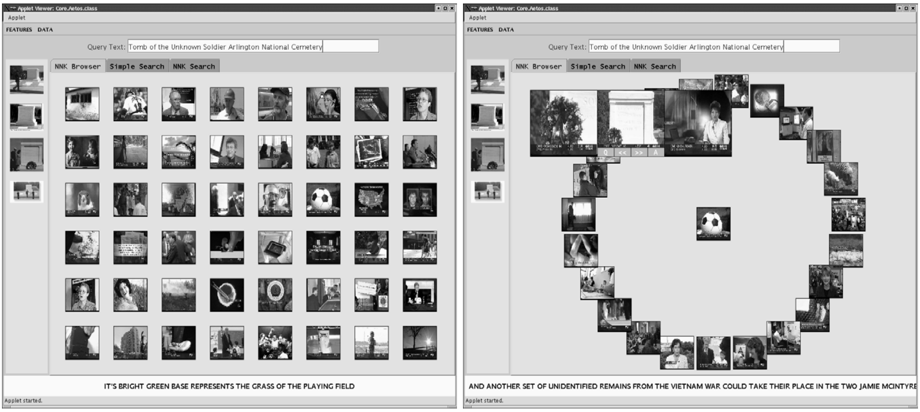
Fig. 3. The display of the first and second step of the NN^k search. The left figure displays the NN^k for the query images shown on the left canvas (planes taking off). The size of each is image proportional to the number of weights for which is was ranked top. The image north-west of the center has been selected as a relevant. The right canvas shows the improved results obtained when querying with the weight set associated with the selected image. See text for details.

outgoing arc in the network. Secondly, image access can be achieved without formulating a query. A mental representation of the image is sufficient to guide the user through the network. Thirdly, the network structure is entirely precomputed which allows interaction to take place in real time, regardless of the size of the collection represented by the network. In addition, we show in [6] that the resulting networks have desirable topological properties, including small-world properties (high clustering coefficient and small average distance between nodes) as defined in [14] and scale-freeness of the vertex degrees, suggesting that it is a structure which lends itself particularly well for browsing.

6.2 Network Visualization and Interaction

To provide initial access points for a user who does not want to formulate a query, we determine the nodes with the highest vertex outdegree. These nodes constitute hubs of the network that allow the user to reach deep into the structure along one or two arcs. The initial display can be seen on the left picture of Figure 4. If the user has formulated a query, we display not the set of high-connectivity nodes as initial access points but the results of the query, that is the same set of images displayed along the spiral in the simple search visualization. Browsing the network is achieved by clicking on any of the displayed nodes. This recovers the set of nearest neighbours from the database, which are then displayed on the screen such that their distances to the center are proportional to their dissimilarity to the selected node (where, again, we use the proportion of weight space for which the NN^k comes top as the measure of similarity).

We shall illustrate the usefulness of the browsing structure using topic 06 asking for images of the "Unknown Soldiers' monument" in Arlington. As the query images show, the monument itself is of white colour, has a very distinctive shape and is set against a relatively dark background. The image from among the high-connectivity nodes that seems visually most similar is the football in the upper left corner on the left picture of Figure 4. Clicking on this image results in the display shown on the right with the football image itself placed in the center and its NN^k displayed around it. The enlarged image in the top right depicts the monument we are looking for (to the left and the right of that NN^k are shown the neighbouring key frames in the corresponding video sequence).



Query Text: Tomb of the Unknown Soldier Arlington National Cemetery

IT'S BRIGHT GREEN BASE REPRESENTS THE GRASS OF THE PLAYING FIELD

Applet started.

Query Text: Tomb of the Unknown Soldier Arlington National Cemetery

AND ANOTHER SET OF UNIDENTIFIED REMAINS FROM THE VIETNAM WAR COULD TAKE THEIR PLACE IN THE TWO JAMIE MCINTYRE

Applet started.

Fig. 4. The left screenshot shows the initial display of high-connectivity hubs of the NN^k network. The right screenshot has the selected image placed in the center and its NN^k arranged around it. Already at this stage we find that one of these images is relevant to the query.

The network structure was used extensively for TRECVID 2003 and proved instrumental for the success of the interactive runs. In one interactive run we restricted interaction to browsing only, and although the performance remained below that of other runs that employed some form of query-by-example, it was comparable to a large number of other interactive runs submitted by the other participants, and was significantly above the performance of our automated search run that used a fixed set of weights without any further user interaction (for details see [4]).

7 Conclusions

We have presented three different techniques for content-based access to image collections and described different visualization for each. There is quantitative evidence suggesting that the two-step NN^k search improves on traditional relevance feedback techniques for weight update, while the NN^k network appears to fare very well as a complement to the traditional query-by-example paradigm. To exploit the full potential of these techniques, however, it is pivotal to tie them to efficient, user-friendly interfaces with a rich set of user interactions. This paper has proposed possible ways how this can be achieved in an integrated retrieval system.

Acknowledgements. This work was partially supported by the EPSRC, UK.

References

1. G Aggarwal, T V Ashwin, and S Ghosal. An image retrieval system with automatic query modification. *IEEE Transactions on multimedia*, 4(2):201–213, 2002.
2. I Campbell. *The ostensive model of developing information-needs.* PhD thesis, University of Glasgow, 2000.
3. M Flickner, H Sawhney, W Niblack, Q H J Ashley, B Dom, M Gorkani, J Hafner, D Lee, D Petkovic, D Steele, and P Yanker. Query by image and video content: the QBIC system. *IEEE Computer*, 9:23–32, 1995.
4. D C Heesch, M Pickering, A Yavlinsky, and S Rüger. Video retrieval within a browsing framework using keyframes. In *Proceedings of TRECVID 2003, NIST (Gaithersburg, MD, Nov 2003)*, 2004.
5. D C Heesch and S Rüger. Performance boosting with three mouse clicks — Relevance feedback for CBIR. In *Proceedings of the European Conference on IR Research 2003*. LNCS, Springer, 2003.
6. D C Heesch and S Rüger. NN^k networks for content based image retrieval. In *Proceedings of the European Conference on IR Research 2004*. LNCS, Springer, 2004.
7. D C Heesch, A Yavlinsky, and S Rüger. Performance comparison between different similarity models for CBIR with relevance feedback. In *Proceedings of the International Conference on video and image retrieval (CIVR 2003), Urbana-Champaign, Illinois*. LNCS, Springer, 2003.
8. Klaus Mainzer. *Computerphilosophie*. Junius Verlag, 2003.
9. H Müller, W Müller, D M Squire, M.S Marchand-Maillet, and T Pun. Strategies for positive and negative relevance feedback in image retrieval. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR 2000), IEEE, Barcelona, Spain*, 2000.
10. T S Rui, T S Huang, M Ortega, and S Mehrota. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 123–131, 1998.
11. S Santini, A Gupta, and R Jain. Emergent semantics through interaction in image databases. *IEEE transactions on knowledge and data engineering*, 13(3):337–351, 2001.
12. Q Tian, B Moghaddam, and T S Huang. Display optimization for image browsing. In *International Workshop on Multimedia Databases and Image Communications*, 2001.
13. R S Torres, C G Silva, C B Medeiros, and H V Rocha. Visual structures for image browsing. In *Conference on Information Knowledge Management (CIKM'03)*, 2003.
14. D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.