NN^k Networks for Content-Based Image Retrieval

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 a novel interaction technique to support content-based image search in large image collections. The idea is to represent each image as a vertex in a directed graph. Given a set of image features, an arc is established between two images if there exists at least one combination of features for which one image is retrieved as the nearest neighbour of the other. Each arc is weighted by the proportion of feature combinations for which the nearest neighour relationship holds. By thus integrating the retrieval results over all possible feature combinations, the resulting network helps expose the semantic richness of images and thus provides an elegant solution to the problem of feature weighting in content-based image retrieval. We give details of the method used for network generation and describe the ways a user can interact with the structure. We also provide an analysis of the network's topology and provide quantitative evidence for the usefulness of the technique.

1 Introduction

The problem of retrieving images based not on associated text but on visual similarity to some query image has received considerable attention throughout the last decade. With its origins in computer vision, early approaches to content-based image retrieval (CBIR) tended to allow for little user interaction but it has by now become clear that CBIR faces a unique set of problems which will remain insurmountable unless the user is granted a more active role. The image collections that are the concern of CBIR are typically too heterogenous for object modelling to be a viable approach. Instead, images are represented by a set of low-level features that are a long way off the actual image meanings. In addition to bridging this semantic gap, CBIR faces the additional problem of determining which of the multiple meaning an image admits to is the one the user is interested in. This ultimately translates into the question of which features should be used and how they should be weighted relative to each other. Relevance feedback has long been hailed as the cure to the problem of image polysemy. Although the performance benefits achieved through relevance feedback are appreciable, there remain clear limitations. One of these is the fast convergence of performance during the first few iterations (e.g. [15], [12]), typically halfway before reaching the global optimum. Also, positive feedback, which turns out to be the

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 253–266, 2004.

most efficient feedback method when the collection contains a sufficiently large number of relevant objects, becomes ineffective if the first set of results does not contain *any* relevant items. Not surprisingly, few papers that report performance gains through relevance feedback use collections of sizes much larger than 1000. Possibly as a response to this limitation, research into the role of negative examples has recently intensified (eg [11], [15]). The general conclusion is that negative examples can be important as they allow the user to move through a collection. [15] concludes that negative feedback "offers many more options to move in feature space and find target images. [...] This flexibility to navigate in feature space is perhaps the most important aspect of a content-based image retrieval system."

We would like to take this conclusion further and claim that in the case of large image collections, it becomes absolutely vital to endow a system with the most efficient structures for browsing as well as retrieval. Relevance feedback on negative examples is arguably one possibility but is relatively inefficient if browsing is a main objective. Motivated by these shortcomings of the traditional query-by-example paradigm and of relevance feedback, this paper proposes a novel network structure that is designed to support image retrieval through browsing. The key idea is to attack polysemy by exposing it. Instead of computing at runtime the set of most similar images under a particular feature regime, we seek to determine the set of images that could potentially be retrieved using any combination of features. We essentially determine the union over all feature combinations of the sets of top ranked images. This is done taking each image of the collection in turn as a query. For each image we store the set of images that were retrieved top under some feature regime and the number of times this happened. The latter number provides us with a measure of similarity between two images. Because nearest neighbourhood need not be reciprocated, the similarity measure is asymmetric and the resulting network a directed graph. We refer to the resulting structure as an NN^k network (NN for nearest neighbour and k for the number of different feature types). As it is entirely precomputed, the network allows interaction to take place in real time regardless of the size of the collection. This is in contrast to query-by-example systems, where the time complexity for retrieval is typically linear in the size of the image collection. The storage requirements for the network increase linearly with the number of images. The time complexity of the network generation algorithm is linear in the number of images and at most quadratic in the number of features. In practice, however, the number of features is constant and, as we will show, does not need to be very large to give respectable results.

Using collections of varying size (238, 6129, 32318), we found that the resulting networks have some interesting properties which suggest that the structures constitute 'small-world' networks [21] at the boundary between randomness and high regularity that should make them ideal for organizing and accessing image collections.

The paper is structured as follows: In section 2, we review work that is related to, or has inspired, the technique here introduced. In section 3, we provide details of how the network structure is generated. Section 4 describes the ways a user

can interact with the browsing structure. Section 5 presents an analysis of the topological properties of the network and section 6 reports on a quantitative performance evaluation of the network. We conclude the paper in section 7.

2 Related Work

The idea of representing text documents in a nearest neighbour network first surfaced in [7]. The network was, however, strictly conceived as an internal representation of the relationships between documents and terms. The idea was taken up in a seminal paper by Cox ([5] and in greater detail in [6]) in which the nearest neighbour network was identified as an ideal structure for interactive browsing. Cox is concerned with structured databases and envisages one nearest neighbour network for each field of the database with individual records allowing for interconnections between the sets of networks.

Notable attempts to introduce the idea of browsing into CBIR include Campbell's work [3]. His ostensive model retains the basic mode of query based retrieval but in addition allows browsing through a dynamically created local tree structure. The query does not need to be formulated explicitly but emerges through the interaction of the user with the image objects. When an image is clicked upon, the system seeks to determine the optimal feature combination given the current query and the query history, i.e. the sequence of past query images. The results are displayed as nodes adjacent to the query image, which can then be selected as the new query. The emphasis is on allowing the system to adjust to changing information needs as the user crawls through the branching tree.

Jain and Santini's "El niño" system [18] and [17] is an attempt to combine query-based search with browsing. The system displays configurations of images in feature space such that the mutual distances between images as computed under the current feature regime are, to a large extent, preserved. Feedback is given similar as in [11] by manually forming clusters of images that appear similar to the user. This in turn results in an altered configuration with, possibly, new images being displayed.

Network structures that have increasingly been used for information visualization and browsing are Pathfinder networks (PFNETs) [8]. PFNETs are constructed by removing redundant edges from a potentially much more complex network. In [9] PFNETs are used to structure the relationships between terms from document abstracts, between document terms and between entire documents. The user interface supports access to the browsing structure through prominently marked high-connectivity nodes. An application of PFNETs to CBIR is found in [4] where PFNETs are constructed and compared with three different classes of image features (colour, layout and texture) using the similarity between images as the edge weight. According to the authors, the principal strength of the network is its ability to expose flaws in the underlying feature extraction algorithm and the scope for interaction is negligible.

What distinguishes our approach from all previous approaches is the rationale underlying and the method used for network generation, as well as a new notion

of similarity between images. In contrast to Cox's networks [5], we determine the nearest neighbour for every combination of features; it is this integration over features that endows the structure with its interesting properties. Also, unlike Pathfinder networks, we do not prune the resulting network but preserve the complete information. This seems justified as we are not concerned with visualizing the entire structure but with facilitating user interaction locally.

3 Network Generation

Given two images X and Y, a set of features, and a vector of feature-specific similarities F, we compute the overall similarity between X and Y as the weighted sum over the feature-specific similarities, i.e.

$$S(X,Y) = \mathbf{w^T F}$$

with the convexity constraint $|\mathbf{w}|_1 = \sum w_i = 1$ and $w_i \geq 0$. Each of the components F_i represent the similarity between X and Y under one specific feature i which itself can be a complex measure such as shape or colour similarity. According to our construction principle, an image X is connected to an image Y by a directed edge $\overrightarrow{XY}$ if and only if Y is the nearest neighbour of X for at least one combination of features, i.e. if and only if there is at least one instantiation of the weight vector $\mathbf{w}$ such that it causes the image Y to have the highest similarity $S(X,Y)$ among all images of the collection (excluding X). Because the overall similarity is a linear sum, small changes in any of the weights will induce correspondingly small changes in the similarity value. Points that are close in weight space should therefore produce a similar ranking and in particular, the same nearest neighbour. We can think of the weight space as being partitioned into a set of regions such that all weights from the same region are associated with the same nearest neighbour. Figure 1 illustrates this idea in the case of a three-dimensional weight space (for details see caption).

The most systematic way of sampling is to impose a grid on the weight space with a fixed number of grid points along each dimension. Using a recursive algorithm with the following recurrence scheme

$$T(1,g) = g$$
$$T(k,g) = \sum_{i=0}^{g} T(k-1, g-i)$$

and setting k and g initially to the number of dimensions and the number of gridpoints along each axis, respectively, we include all permissible gridpoints.

According to this sampling scheme, an image could have more than one thousand nearest neighbours using five features and a grid size of 0.1. In practice, however, the number of distinct neighbours is much smaller and rarely exceeds 50.

The resolution of the grid that is required to capture all nearest neighbours, therefore, is relatively low. Moreover, lacking any additional information, a nearest neighbour that corresponds to a large volume in weight space may reasonably

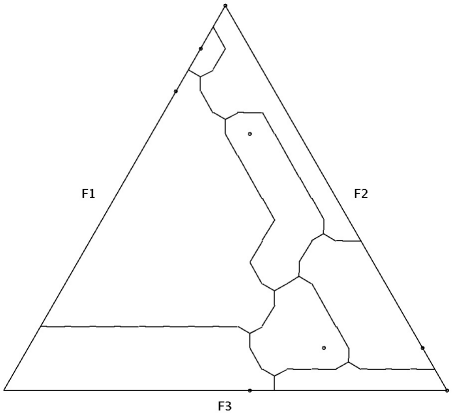

Fig. 1. Simplex showing the partitioning of the weight space into distinct regions for one particular query image. The weights of each of the three features $F1$, $F2$ and $F3$ increase with distance to the corresponding base of the triangle. Each of the bounded regions comprise all those weight sets for which the query has the same nearest neighbour. The points denote the weight combination for each region for which the nearest neighbour had minimum distance to the query.

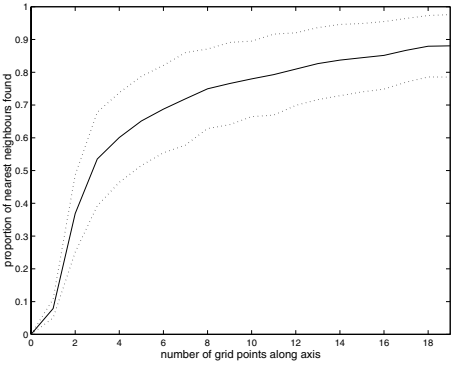

Fig. 2. The proportion of nearest neighbours found for a given grid size averaged over fifty queries (dotted lines: one standard deviation). The exact number of nearest neighbours (100%) for a given query is estimated using 100 gridpoints along each dimension.

be considered more important than one the grid search misses. Figure 2 shows how the number of nearest neighbours rapidly approaches the exact number as the grid size becomes smaller.

It is important to stress, that although, technically, the number of sampled grid points grows exponentially with the dimensionality of the weight space, i.e. the number of features, in practice this number is fixed and limited. Few CBIR applications use more than 10 features. As an illustration, using 7 features and a grid size of 5 per axis, we have a total of 210 grid points to sample.

Using a collection of 32000 images, this can be done in around 50 hours on a standard home computer. With more sophisticated sampling algorithms (such as hierarchical refinement sampling) and parallelization, network construction should be no performance bottleneck even for high-dimensional feature spaces.

For each image we store the set of its nearest neighbours. For each nearest neighbour we also store the proportion of feature combinations in the grid for which that image was ranked top. This number becomes our measure of similarity between two images.

4 Accessing the Network

In order to allow searches without without formulating any query, we provide the user with a representative set of images from the collection by clustering high-connectivity nodes and their neighbours up to a certain depth. Clustering is achieved using the Markov chain clustering (MCL) algorithm [20]. The algorithm reduces the adjacency matrix of the directed graph to a stochastic matrix whose entries can be interpreted as the transition probabilities of moving from one image to another. These probabilities are iteratively updated through an alternating sequence of matrix multiplications and matrix expansions, which have the effect of strengthening frequently used edges. The algorithm has robust convergence properties and allows one to specify the granularity of the clustering. The clustering can be performed offline and may therefore involve the entire image collection. The high sparsity of the adjacency matrix makes the MCL algorithm suitable for even very large networks using sparse matrix techniques.

The interface with the clustering result is shown in Figure 3. We aim to minimize overlap between images while at the same time preserving the cluster structure. The user may select any of the images as a query or as the entry point into the network. Clicking on an image moves it into the center and results in a display of its nearest neighbours. If the size of the set is above a certain threshold the actual number of images displayed is reduced. This threshold T depends on the current size of the window and is updated upon resizing the window. This adjustment is desirable in order to be able to accommodate different screen sizes and makes the system work gracefully with networks that have large variability in the connectivity of its constituent nodes. The criterion for removing images from the set of nearest neighbours is the weight of the arc by which it is connected to the central image (i.e. the area in weight space for which this image is top ranked), only the T images with the highest edge weights are displayed. The neighbours are displayed such that their distances to the central node is a measure of the strength of the connecting edges. The arrangement is found by simulating the evolution of a physical network with elastic springs connecting adjacent nodes.

Through the set of buttons at the bottom of each image, the user can either add images to a query panel (Q) positioned on the left hand side of the display (these images can then be used to start a traditional query-by-example run on the collection), or collect interesting images on a separate panel (A).

Fig. 3. Initial interface in browsing mode. Displayed are the clusters as determined by the Markov Chain Clustering algorithm. Images become larger when moving the mouse over them.

5 Topological Analysis

5.1 Small-World Properties

An interesting and significant feature of the resulting structure is the presence of so-called small-world properties [21]. Small-world graphs are characterized by two topological properties, both of which are relevant in the context of information retrieval: (i) the clustering coefficient and (ii) the average distance between nodes.

Following Watts and Strogatz [21], one of the basic properties of graphs is the clustering coefficient C. It measures the extent to which a vertex' neighbours are themselves neighbours. More formally, given a graph G without loops and multiple edges and a vertex v, the local clustering coefficient at v is given by the ratio of the number of edges between neighbours of v and the maximum number of such edges (given by $\binom{d_G(v)}{2}$ where $d_G(v)$ is the vertex outdegree of v in G). The clustering coefficient is then obtained by averaging the local clustering coefficient over all vertices. We can think of the clustering coefficient as a measure of the randomness of the graph. It attains a maximum in regular lattice graphs and decreases as we replace edges in the regular graph by randomly positioned edges ([21], [13]). A high clustering coefficient seems, prima facie, to be best suited for the task of information retrieval. However, the

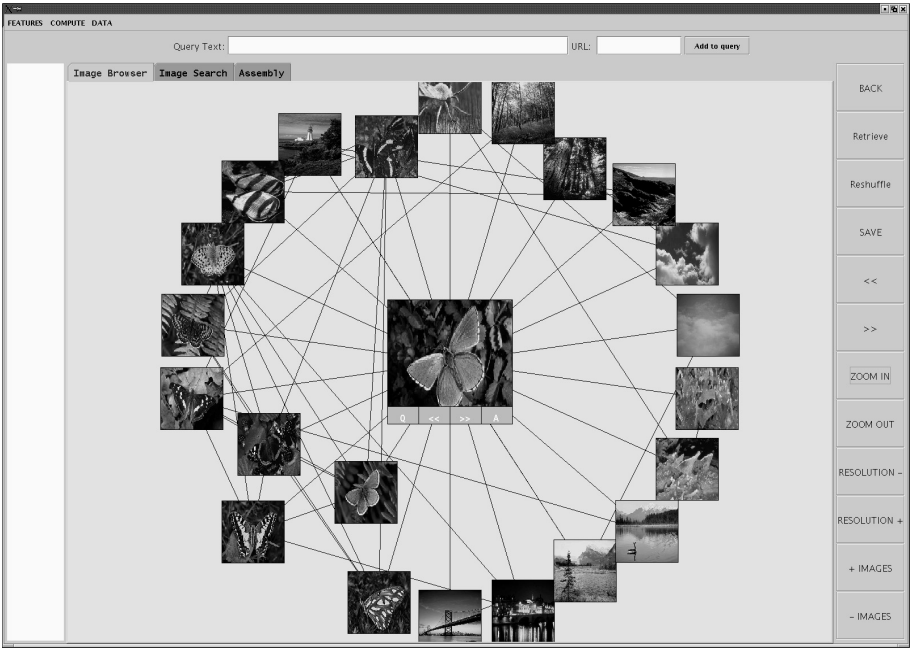

Fig. 4. Local network around the chosen butterfly image depicted in the centre

more organized the structure the more difficult it becomes to efficiently move to different areas of the network. Moreover, the simple organization principle that underlies a lattice graph seems inadequate to capture the semantic richness and ambiguity of images. For the purpose of information retrieval, therefore, it appears desirable to have the information organized in structures that are inbetween the two extremes of regularity and randomness.

We have evaluated the clustering coefficients and average distances for three different collections with different feature sets and sizes varying from 238 to 32,318 images (= number of vertices in the network). The clustering coefficient can easily be compared to what would be expected for a random graph. For the classic Erdös-Rényi graph, the expected clustering coefficient is given by z/n where z is the average vertex degree of a graph with n vertices [16]. Likewise, the average distance in a random graph can be approximated by $l = \log(n)/\log(z)$ with n and z as before [2]. For all the three collections examined, the path length is very close to the result of the random graph model while the clustering coefficient exceeds the predicted value by magnitudes, suggesting that the network has indeed a high degree of local structure. The results are summarized in Table 1.

5.2 Degree Distribution

It is of particular interest to see whether the vertex degree sequence is scale-invariant. A large number of distributed systems from social over communication to biological networks display a power-law distribution in their node degree,

Table 1. Analysis of network structure for three different collections. $C(G)$ and $C_{rand}(G)$ denote the clustering coefficients for, respectively, the actual network and a random network with the same number of vertices and edges. The diameter is the largest distance between any two vertices and thus provides an additional measure of the graph's connectivity.

	Collection		
	Corel	Sketches	Video
Features	5.0	4.0	7.0
Vertices (n)	6,192.0	238.0	32,318.0
Edges (e)	150,776.0	1,822.0	1,253,076.0
Avg Vertex Degree (z)	24.35	7.69	38.77
$C(G)$	0.047	0.134	0.14
$C_{rand}(G)$	0.004	0.03	0.0012
Avg Dist	3.22	3.29	3.33
Avg Dist (rand)	2.73	2.68	2.83
Diameter	6.0	7.0	6.0

reflecting the existence of a few nodes with very high degree and many nodes with low degree, a feature which is absent in random graphs. While initial work on scale-free graphs was concerned with investigating their properties and developing generative models, an issue which has only very recently been looked at and which is of relevance to CBIR is the problem of search in these networks when little information is available about the location of the target [1]. Analysis of the degree distributions of the directed graphs constructed thus far suggests that they are, across a broad range of node degrees, scale-free. Figure 5 depicts the frequency distribution of the in-degrees for the network of the video key frame collection (32,318 images). Note that we use the log-scale along the y-axis. If the relationship were of the form $y = e^{-ax+b}$ and thus corresponded to a power-law distribution, the logarithmic plot would reveal this as a straight line $\ln y = -ax + b$. It is typical for such distributions that their boundedness on one or both sides cause the power-law relationship to break down at the boundaries. So in this case, where the number of nodes with exceedingly few neighbours is in fact very small. For a large range of node degrees, however, the relative frequencies seem fairly well described by a power-law distribution.

6 TRECVID 2003 Evaluation

TRECVID (previously the video track of TREC) provides a rare opportunity for research groups in content-based video retrieval to obtain quantitative performance results for realistic search tasks and large image collections. The search task in 2003 involved 24 topics, each exemplified by a set of images and a short text. For each topic, the task was to find the most similar shots and to submit a list with the top ranked 1000 images. Any type of user interaction was allowed after the first retrieval but time for the search was limited to 15 minutes for each topic.

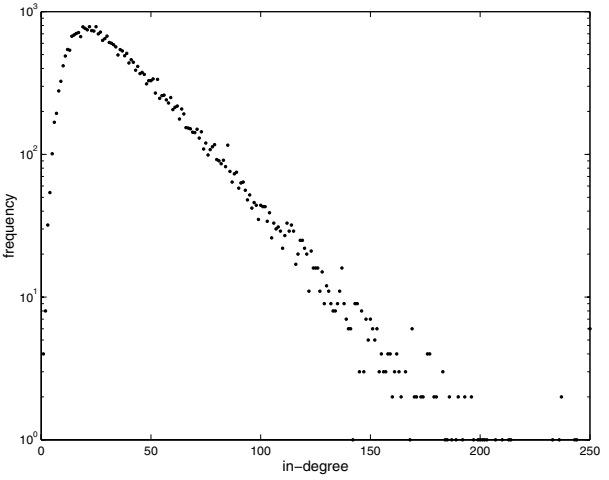

Fig. 5. In-degree distribution for the NN^k network constructed for the video key frame collection

6.1 Features

The NN^k network for the search collection was constructed using seven low-level colour and texture features as well as text from the video transcripts. For the simple texture features, we decided to partition the images into tiles and obtain features from each tile individually with the aim of better capturing local information. The final feature vector for these features consisted of a concatenation of the feature vector of the individual tiles. What follows is a detailed description of each of the features.

HSV Global Colour Histograms: HSV is a cylindrical colour space with H (hue) being the angular, S (saturation) the radial and V (brightness) the height component. This brings about the mathematical disadvantage that hue is discontinuous with respect to RGB coordinates and that hue is singular at the achromatic axis $r = g = b$ or $s = 0$. As a consequence we merge, for each brightness subdivision separately, all pie-shaped 3-d HSV bins which contain or border $s = 0$. The merged cylindrical bins around the achromatic axis describe the grey values which appear in a colour image and take care of the hue singularity at $s = 0$. Saturation is essentially singular at the black point in the HSV model. Hence, a small RGB ball around black should be mapped into the bin corresponding to $hsv = (0, 0, 0)$, to avoid jumps in the saturation from 0 to its maximum of 1 when varying the singular RGB point infinitesimally. There are several possibilities for a natural subdivision of the hue, saturation and brightness axes; they can be subdivided i) linearly, ii) so that the geometric volumes are constant in the cylinder and iii) so that the volumes of the nonlinear transformed RGB colour space are nearly constant. The latter refers to the property

that few RGB pixels map onto a small dark V band but many more to a bright V interval of the same size; this is sometimes called the HSV cone in the literature. We use the HSV model with a linear subdivision into 10 hues, 5 saturation values and 5 V values yielding a 205-dimensional feature vector. The HSV colour histogram is normalised so that the components add up to 1.

Colour Structure Descriptor: This feature is based on the HMMD (hue, min, max, diff) colour space and is part of the MPEG-7 standard [14]. The HMMD space is derived from the HSV and RGB spaces. The hue component is the same as in the HSV space, and max and min denote the maximum and minimum among the R, G, and B values, respectively. The diff component is defined as the difference between max and min. Following the MPEG-7 standard, we quantise the HMMD non-uniformly into 184 bins with the three dimensions being hue, sum and diff (sum being defined as $(max + min)/2$) and use a global histogram.

In order to capture local image structure, we slide a 8×8 structuring window over the image. Each of the 184 bins of the colour structure histogram contains the number of window positions for which there is at least one pixel falling into the corresponding HMMD bin. This descriptor is capable of discriminating between images that have the same global colour distribution but different local colour structures. Although the number of samples in the 8×8 structuring window is kept constant (64), the spatial extent of the window differs depending on the size of the image. Thus, for larger images appropriate sub-sampling is employed to keep the total number of samples per image roughly constant. The 184 bin values are normalised by dividing by the number of locations of the structuring window; each of the bin values falls thus in the range $[0, 1]$, but the sum of the bin values can take any value up to 64 (see [14] for details).

Thumbnail feature: This feature is obtained by scaling down the original image to 44×27 pixels and then recording the gray value of each of the pixels leaving us with a feature vector of size 1,188. It is suited to identify near-identical copies of images, eg, key frames of repeated shots such as adverts.

Convolution filters: For this feature we use Tieu and Viola's method [19], which relies on a large number of highly selective features. The feature generation process is based on a set of 25 primitive filters, which are applied to the gray level image to generate 25 different feature maps. Each of these feature maps is rectified and downsampled and subsequently fed to each of the 25 filters again to give 625 feature maps. The process is repeated a third time before each feature map is summed to give 15,625 feature values. The idea behind the three stage process is that each level 'discovers' arrangements of features in the previous level and ultimately leads to a set of very selective features, each of which takes high values only for a small fraction of the image collection. The feature generation process is computationally quite costly, but only needs to be done once.

Variance Feature: The variance feature is a 20 bin histogram of gray value standard deviations within a a sliding window of size 5×5 determined for each window position. The histogram is computed for each of 9 non-overlapping image tiles and the bin frequencies concatenated to give a feature vector of size 180.

Uniformity Feature: Uniformity is another statistical texture feature defined as

$$U := \sum_{z=0}^{L-1} p^2(z)$$

where $L = 100$ is the number of gray levels and $p(z)$ the frequency of pixels of gray level z. For each of 8×8 image tiles, we obtain one uniformity value resulting in a feature vector of size 64.

Bag of words: Using the textual annotation obtained from the video transcripts provided, we compute a bag-of-words feature consisting for each image of the set of accompanying stemmed words (Porter's algorithm) and their weights. These weights are determined using the standard tf-idf formula and normalised so that they sum to one. As this is a sparse vector of considerable size (the number of different words) we store this feature in the form of (weight, word-id) pairs, sorted by word-id.

6.2 Distances and Normalisation

In order to compare two images in the collection we use distances of their corresponding features. For these we use the L_1-norm throughout (and the L_1 norm raised to the power of 3 for the bag-of-stemmed-words). Some of the distances already exhibit a natural normalisation, for example when the underlying features are normalised (eg the HSV colour histograms), others do not (eg the colour structure descriptor). As the distances for different features are to be combined, we normalise the distances empirically for each feature, such that their median comes to lie around one.

6.3 Results

Four interactive runs were carried out, in one of which the user was only allowed to find relevant shots by browsing through the NN^k network. For this run text and images were only used to inform about the task. Although our best interactive runs were those that employ a mixture of search, relevance feedback and browsing, the performance (as measured in mean average precision over all 24 topics) of the browsing-only run was considerably better than that of a manual run in which images were retrieved with a fixed set of feature weights and no subsequent user interaction. Performance also proved superior to more than 25% of all the 36 interactive runs submitted by the participating groups, all of which used some form of automatic search-by-exammple. Considering the number of

Table 2. Performance of the browse-only run compared to our interactive search run with browsing and our best manual run with no user interaction and the mean and median of the 36 interactive runs from all groups.

	Mean Average Precision
TRECVID Median	0.1939
TRECVID Mean	0.182 ± 0.088
Search + Browsing	0.257 ± 0.219
Browsing only	0.132 ± 0.187
Manual Run	0.076 ± 0.0937

features and the size of the collection, these results are quite respectable and demonstrate that browsing in general and the proposed structure in particular have a potential for CBIR that should not be left unexploited. A summary of the results is given in Table 2 and more details can be found in [10].

7 Conclusions

The strengths of the proposed structure are twofold: (i) it provides a means to expose the semantic richness of images and thus helps to alleviate the problem of image polysemy which has been for many years a central research concern in CBIR. Instead of displaying all objects that are similar under only one, possibly suboptimal, feature regime, the user is given a choice between a diverse set of images, each of which is highly similar under *some* interpretation, (ii) the structure is precomputed and thus circumvents the often inacceptable search times encountered in traditional content-based retrieval systems. Interaction is in real time, regardless of the collection size.

The NN^k technique presented here is of wider applicability. Its usefulness naturally extends to any multimedia objects for which we can define a similarity metric and a multidimenional feature space, such as text documents or pieces of music. It is, however, in the area of image retrieval that it should find its most profitable application as relevance can be assessed quickly and objects can be displayed at a relatively small scale without impeding object understanding. Although the principal motivation behind the NN^k network is to mitigate the problems associated with category search in large collections, the topology should make it an ideal structure for undirected browsing also.

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

References

1. L A Adamic, R M Lukose, A R Puniyani, and B A Huberman. Search in power-law networks. *Physical Review E*, 64, 2001.
2. B Bollobás. *Random Graphs.* Springer, New York, 1985.

3. I Campbell. *The ostensive model of developing information-needs.* PhD thesis, University of Glasgow, 2000.
4. C Chen, G Gagaudakis, and P Rosin. Similarity-based image browsing. In *Proceedings of the 16th IFIP World Computer Congress. International Conference on Intelligent Information Processing*, 2000.
5. K Cox. Information retrieval by browsing. In *Proceedings of The 5th International Conference on New Information Technology, Hongkong*, 1992.
6. K Cox. *Searching through browsing.* PhD thesis, University of Canberra, 1995.
7. B Croft and T J Parenty. Comparison of a network structure and a database system used for document retrieval. *Information Systems*, 10:377–390, 1985.
8. D W Dearholt and R W Schvaneveldt. *Properties of Pathfinder networks, In R W Schvaneveldt (Ed.), Pathfinder associative networks: Studies in knowledge organization.* Norwood, NJ: Ablex, 1990.
9. R H Fowler, B Wilson, and W A L Fowler. Information navigator: An information system using associative networks for display and retrieval. *Department of Computer Science, Technical Report NAG9-551, 92-1*, 1992.
10. D Heesch, M Pickering, A Yavlinsky, and S Rüger. Video retrieval within a browsing framework using keyframe. In *Proceedings of TRECVID 2003, NIST (Gaithersburg, MD, Nov 2003)*, 2004.
11. 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.
12. 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.
13. J M Kleinberg. Navigation in a small world. *Nature*, page 845, 2000.
14. B S Manjunath and J-S Ohm. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11:703–715, 2001.
15. 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.
16. M E J Newman. *Random graphs as models of networks, In S Bornholdt and H G Schuster (Ed.), Handbook of graphs and networks - from the genome to the internet.* Wiley-VCH, 2003.
17. 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.
18. Simone Santini and Ramesh Jain. Integrated browsing and querying for image databases. *IEEE MultiMedia*, 7(3):26–39, 2000.
19. Kinh Tieu and Paul Viola. Boosting image retrieval. In *5th International Conference on Spoken Language Processing*, December 2000.
20. S van Dongen. A cluster algorithm for graphs. *Technical report, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam*, 2000.
21. D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.