Multimed Tools Appl (2008) 40:261–284
DOI 10.1007/s11042-008-0207-2

A survey of browsing models for content based image retrieval

Daniel Heesch

Published online: 22 April 2008

Abstract The problem of content based image retrieval (CBIR) has traditionally been investigated within a framework that emphasises the explicit formulation of a query: users initiate an automated search for relevant images by submitting an image or draw a sketch that exemplifies their information need. Often, relevance feedback is incorporated as a post-retrieval step for optimising the way evidence from different visual features is combined. While this sustained methodological focus has helped CBIR to mature, it has also brought out its limitations more clearly: There is often little support for exploratory search and scaling to very large collections is problematic. Moreover, the assumption that users are always able to formulate an appropriate query is questionable. An effective, albeit much less studied, method of accessing image collections based on visual content is that of browsing. The aim of this survey paper is to provide a structured overview of the different models that have been explored over the last one to two decades, to highlight the particular challenges of the browsing approach and to focus attention on a few interesting issues that warrant more intense research.

Keywords Image retrieval · CBIR · Human-computer interaction · Data visualization · Browsing · Networks · Clustering · Dimensionality reduction

1 Introduction

With the advent of the information age and the concomitant explosion of digital data in the form of texts, music, images and videos, the question of how to retrieve relevant information from potentially very large repositories has become of immense

D. Heesch (✉)
Department of Electrical and Electronic Engineering,
Imperial College London, SW7 2AZ, London, UK
e-mail: daniel.heesch@imperial.ac.uk

practical importance. The ability to retrieve stored items from memory based on similarity has been recognised as a key property that underlies much of what we associate with human intelligence including analogical reasoning, classification and prediction [90]. Yet, what we humans solve effortlessly remains a formidably difficult task for a machine. At the core of the problem lies the deceptively simple notion of similarity. In essence, the difficulty arises from the fact that not all features by which objects may be represented are equally useful for measuring similarity. The notion of similarity predicates a distinction between essential and accidental features. For example, an essential feature for something to be a bike is the possession of two free-running wheels linked by a chain, while the colour is accidental. Essential features are thus those that are shared by all individuals of a class. When comparing objects within the same class, similarity is presumably based on accidental features, while similarity between objects of different classes tends to be judged by differences in their essential properties (e.g. a three-wheel cart is judged more similar to a bike than a four-wheel car). Being able to discriminate between features that matter and those that don't is a major learning task and requires substantial training and possibly causal theories of the domain [81] for a review on the relationship between similarity and categorisation see [77].

The problem of estimating the relative significance of different features pertains to information retrieval in general. It is however greatly compounded in the case of image retrieval in two significant ways: First, while documents readily suggest a representation in terms of their constituent words, images do not generally admit to such a natural decomposition into semantic atoms. This renders image representations to some extent arbitrary. Secondly, images typically admit to a multitude of different meanings, each of which may have its own set of supporting visual features.

Content based image retrieval (CBIR) inherited its early methodological focus from the by then already mature field of text retrieval. The primary role of the user is that of formulating a query, while the system is given the task of finding relevant matches. The spirit of the time is well captured in Gupta and Jain's classic review paper from 1997 [33] in which they remark that "an information retrieval system is expected to [...] help a user specify an expressive query to locate relevant information." By far the most commonly adopted method for specifying a query is to supply an example image (known as query by example or QBE), but other ways have been explored. Recent progress in automated image annotation, for example, reduces the problem of image retrieval to that of standard text retrieval with users merely entering search terms. Whether this makes query formulation more intuitive for the user remains to be seen. In other systems, users are able to draw rough sketches possibly by selecting and combining visual primitives, e.g. [42, 79] and [25]. All these methods have in common that at some point users issue an explicit query, be it textual or pictorial.

This division of roles between the human and the computer system as exemplified by many early CBIR systems seems warranted on the grounds that search is not only computationally expensive for large collections but also amenable to automation. However, when one considers that humans are still far better at judging relevance, and can do so rapidly, the role of the user seems unduly curtailed. The introduction of relevance feedback into image retrieval has been an attempt to involve the user more actively and has turned the problem of learning feature weights into a supervised learning problem (for reviews see [21, 100] and [37]). Although the incorporation

of relevance feedback techniques can result in substantial performance gains, such methods fail to address a number of important issues. Users may, for example, not have a well-defined information need in the first place and may simply wish to explore the image collection. Should a concrete information need exist, users are unlikely to have a query image at their disposal to express it. Moreover, nearest neighbour search requires efficient indexing structures that do not degrade to linear complexity with a large number of dimensions [88].

Several years after Gupta and Jain's paper, voices from the computer vision community began to urge for a broadening of the research programme. For example, Forsyth [27] criticised that "[...] search has been overemphasized by the content based image retrieval literature, and a number of other interesting activities—browsing, organising and image data mining have not been sufficiently well studied."

Browsing provides an interesting alternative to systems requiring explicit query formulation, but it has, by comparison, received only scant attention. A number of definitions of browsing have been suggested in the literature. Spence [80] uses the term to imply a scanning activity that allows users to build a cognitive map of a domain. A classic example of browsing according to him is the scanning of articles on the front cover of a newspaper. In [10], the authors distinguish between three kinds of browsing: "scan-browse", "review-browse" and "search-browse". Our use of the term is most akin to the last one. We here define browsing as the exploration of potentially very large spaces through a sequence of local decisions or navigational choices. Note that there is a body of research that is concerned with browsing *individual* images, mostly from remote sensing applications (e.g. [23] and [7]). This line of research can be identified with the "scan-browse" activity which is not the concern of our paper.

Most of the browsing models to be discussed cast the collection into a structure that can be navigated interactively. Arguably one of the greatest difficulties of a browsing approach is to identify structures that are conducive to effective search in the sense that they support fast navigation, provide a meaningful neighbourhood for choosing a browsing path and allow users to position themselves in an area of interest [18].

Overview papers on content based image search tend to cover the QBE methodology well but have little to say about browsing, including the otherwise exemplary work by [78] and the more recent follow-up study [31]. The epically sized and even more recent survey paper by [22] purports to have wide coverage but has a clear focus on techniques for automated image annotation. The principal objective of this paper is to fill the gap by providing an extensive survey and analysis of current browsing methods, and to stimulate further, and more concerted research in this area.

The paper has the following structure: Section 2 puts forth four reasons why browsing should be considered as an interesting access method. Section 3 introduces the principal challenges of browsing models and subsequently investigates three different classes of browsing structures: the first two consist of static structures that have been precomputed prior to user interaction and we shall distinguish between hierarchies (Section 3.1) and networks (Section 3.2). In Section 3.3, we shall look at more dynamic models that incorporate some form of user feedback and tend to employ a mixture of the techniques described in the previous two sections. Finally in Section 4, we shall discuss the merits of different approaches and motivate directions for future research. Section 5 ends the paper with a short conclusion.

2 In defense of browsing

Mental query The query in CBIR often takes the form of an example image. This query mode is inadequate when query images are not readily at hand. Indeed, users would perhaps need to access a collection first by some other means to identify suitable query images (for example through browsing as advocated in [76]). Early systems have attempted to overcome this limitation by allowing users to draw sketches. This approach has limited expressive power and, with traditional input devices, is operationally cumbersome. Another solution is automated image annotation [26, 49, 94, 99] which promises to reduce visual search methodologically to traditional text retrieval. However, images, whether in our mind or not, can be inordinately more expressive than words and to find what lies beyond verbal description, a visually guided search is likely to remain the more effective strategy. Browsing, meanwhile, can be accomplished with only a mental representation of the query as a guide, and that query may be as simple or as complex as we wish.

Fluid information needs Retrieving images based on an explicit query requires users to have a concrete information need in the first place. This may often not be the case. Instead, the information need may initially be very vague and develop in the course of and as a result of the interaction with the database. Gaining an overview of a collection and being able to navigate quickly between different kinds of images then becomes crucial. Depending on how the collection is structured, browsing can provide much greater support for undirected search and help users develop information needs.

Exploiting the cognitive abilities of users The ability of the human visual system to recognise patterns reliably and quickly is a marvel yet to be fully comprehended. Endowing systems with similar capabilities has proven an exceedingly ambitious task. Early approaches towards content based retrieval left to a large extent unexploited the cognitive prowess of the user. The problem was understood as one of computation and representation, not one of human-computer interaction. Given our present limitations in understanding and emulating cognitive vision, however, the most promising way to leverage the potential of computers is to combine their strengths with those of users and to achieve a synergy through interaction.

Such synergy can be achieved through browsing as users are continuously required to make decisions based on the relevance of items in relation to their current information need. Since humans are much better at recognising whether something is relevant than to describe what an item has to look like to be considered relevant, a substantial amount of time is spent by engaging users in what they are best at, while exploiting computational resources to render interaction fast.

Responsiveness Much research in content based retrieval is aimed at improving retrieval performance. Comparatively little effort has been directed towards improving the scalability properties of retrieval methods. In the case of a visual query, image retrieval entails a nearest neighbour search amongst a database of images, or the feature representations thereof. For low-dimensional feature spaces, tree-based indexing structures that partition the data space hierarchically such as KD-trees [4], R-trees [34], R*-trees [3] or SR-trees [43] improve considerably over a sequential scan of the database. However, for the high-dimensional spaces that are typical of

multimedia applications, the advantage of these structures breaks down as has been shown both theoretically and empirically [6, 86, 88]. Vector approximation files [87] have been proposed as an alternative to these hierarchical structures. They consist of a low-dimensional approximation of the original feature vector and can be used to efficiently discard the majority of dissimilar objects in a first run. While alleviating the curse of dimensionality that affects partitioning approaches, supporting efficient database searches of millions of multimedia objects continues to be a challenge even with these more recent proposals. Consider now the task of browsing the World Wide Web (WWW). The speed with which the hyperlinked network of sites can be browsed is clearly independent of the size of the WWW, and more or less independent of the sites that are linked together. The neighbours are already there before users hit a particular page. In the case of the WWW, the structure has evolved over many years and links have been established manually. Yet, the example illustrates the fact that, once built, browsing structures can be navigated very quickly.

3 Structuring data

The advantages of browsing outlined in the previous section come at a price. Arguably the greatest challenge is to identify good organisation principles for structuring a collection. Intuitively, we should wish objects to be near each other and easily accessible from one another if they are similar. As argued earlier, the notion of similarity is fraught with difficulties. Images admit to a number of different representations in terms of visual features and not all features are equally useful to find, for a particular image, those that are similar. The question of how to weigh different features when constructing structures for browsing is far from trivial. In addition, we have a choice between different topologies and some are better suited than others depending on the application. Hierarchical organisations may be well suited when there is an obvious dimension along which to arrange and split the collection of objects. When there are multiple ways in which objects can be related to each other (e.g. sites on the WWW, where links can carry very different semantics), networks might prove more useful.

We shall distinguish between three classes: (i) static hierarchical structures, (ii) static networks and (iii) dynamic structures. We end each of the three sections with a summary and discussion.

3.1 Static hierarchies

Hierarchies have a universal presence in our daily lives: examples include the organisation of files on a computer, the arrangement of books in a physical library, the presentation of information on the web, organisational structures, postal addresses and many more.

Hierarchical structures have been studied for many years as a possible remedy against the linear time complexity of exhaustive nearest neighbour searches [29] with recent applications to image search [47, 53, 60, 66, 89]. The general idea here is to find nearest neighbours by descending a tree of hierarchically organised cluster centroids. At every step, a comparison is performed between the representation of the query and the representation stored at the given node.

In an interactive setting, it is the users who at every step compare their internal query image with the cluster centroids at a particular level of the hierarchy and decide along which path to continue. To use hierarchical structures for browsing, users need to be able to reliably predict in which of the multiple subtrees the target image resides. When it is obvious along which dimension to hierarchically split the database, the navigational choice becomes relatively simple. However, such cannot be said for the low-level, high-dimensional feature representations commonly used for image data, so choosing the 'right' hierarchy is a challenge.

The most common methods for constructing hierarchies is by way of clustering (see [24] for a general text and [5] for a comprehensive recent survey). Note that this paper does not aim to provide an overview of clustering techniques in general. Instead the presentation is restricted to those that have been employed in the context of CBIR.

A useful distinction is between hard clustering and fuzzy or overlapping clustering models (e.g. [97]). In the former, an item is unambiguously assigned to only one cluster, while in the latter an item may belong to several clusters. We treat each of these two in turn.

3.1.1 Hard clustering

Agglomerative methods Hard clusters can be obtained either by iteratively merging clusters (agglomerative clustering) or by recursively partitioning clusters (divisive clustering). Early applications of agglomerative clustering to image browsing are described in [95, 96, 98] and [46]. The first two papers are concerned with video browsing. Clustering here involves automated detection of topics and for each topic the detection of constituent stories. Stories are represented as video posters, a set of images from the sequences that act as pictorial summaries. In [96], the authors cluster video shots and build "scene transition graphs" that visualise story development in videos with particular application to story-based movies (rather than topic-based news videos).

In [98] and [91] the self-organising map (SOM) algorithm [45] is applied to map images onto a two-dimensional grid. The resulting grid is subsequently clustered agglomeratively. More recent applications of the algorithms are by [48] where images are mapped to a hierarchically organised SOM for each of a set of three visual features. Based on relevance feedback, the images of the different SOMs are merged. One of the major drawbacks of the SOM algorithm (and neural network architectures in general) is its computational complexity. Training instances often need to be presented multiple times and convergence has to be slow in order to achieve good performance, in particular for dense features (see [65] for a scalable map applied to sparse document representations).

In [14] and [15], Chen et al. propose the similarity pyramid to represent image collections. Each level is organised such that similar images are in close proximity on a two-dimensional grid. Images are first organised into a binary tree through agglomerative clustering based on pairwise similarities. The binary tree is subsequently transformed into a tree in which each node has four instead of only two children, a datastructure that is technically known as a quadtree (from *quadrant*). A quadtree not only provides users with a greater choice at each level of the hierarchy, it also makes better use of the two-dimensional display space. Cluster representatives are arranged such that some measure of overall visual coherence is maximised. Since

the quadtree structure is precomputed, the computational cost incurred at browsing time is slight. Figure 1 illustrates the process of conversion from a binary tree (left) into a quadtree (right). Considering only the root level of the binary tree, the goal is to choose four nodes from the fourteen such that their subtrees contain all the leaf nodes {8..15}. While this may always be achieved by selecting the nodes two levels deeper, i.e. {4, 5, 6, 7}, it is observed in [15] that a mixture of nodes from different levels can provide visually more balanced representations.

Essential to most clustering methods is the notion of a distance between objects. In many applications it is not obvious which distance function should be used or whether this function should even possess metric properties. Even if one may intuit an appropriate functional form, the instantiation of parameters, such as the weights associated with different visual features, remains a problem. Goldberger et al. [32] propose one solution to this problem of choice. The distance measure is obtained from the information bottleneck (IB) method of Tishby et al. [83]: given a set of objects X and their properties Y, the IB method formulates the problem of clustering as one of finding a mapping from X to a smaller set $\hat{X}$ of fixed size such that the predictions of Y from X through $\hat{X}$ are as close as possible to the direct prediction of Y from X. This can be measured in terms of the difference in mutual information, $d(x_1, x_2) = I(X, Y) - I(\hat{X}, Y)$ where x_1 and x_2 are the two clusters being merged to get from X to $\hat{X}$. At each step of the agglomerative clustering procedure, the two clusters to be merged are chosen such that the decrease in mutual information is minimised. The results reported in [32] compare favourably with a metric based merging criterion on the same data set used in [46].

Although agglomerative clustering is quadratic in the number of images, approximation methods can reduce the computation time considerably. An example is the approach by [15] which combines inexact but fast nearest neighbour search with sparse distance matrices.

Divisive methods A computationally more attractive alternative is divisive clustering whereby clusters are recursively split into smaller clusters. One popular clustering algorithm for this purpose is k-means. In [61], this algorithm is applied to 6,000 images with cluster centroids being displayed according to their position on a global Sammon map (see Section 3.3). Compared to agglomerative clustering, the divisive approach has been found to generate less intuitive groupings (e.g. [15, 96]) and the former has remained the method of choice in spite of its computational complexity.

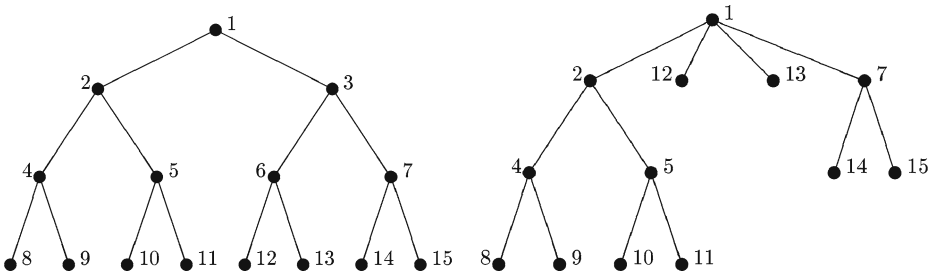

Fig. 1 A binary tree (*left*) and the first step towards converting it into a quadtree (*right*), in which each node will have four children

The pairwise similarities between images can be construed as edge weights of a complete graph in which vertices correspond to images. It is then possible to segment the graph by removing those edges the weights of which lie below a certain threshold. Cheung and Zakhor [16] uses such a method to identify near-identical sequences in video databases. The distance relationships between sequences are represented as an undirected graph in which two sequences are connected if their distance is below some threshold. For a suitable threshold, the graph partitions into a number of connected components. Of these only those with an edge density above a user-set threshold are recognised as clusters and are removed from the graph. By successively removing the remaining edges in decreasing order of length, new clusters satisfying the density criterion may subsequently be found. Note that a thresholded graph is distinct from the notion of a *threshold graph*, which has a very different meaning in graph theory.

3.1.2 Fuzzy clustering

Common to the methods discussed so far is their inability to model membership to multiple clusters or to provide a measure of confidence of a particular cluster assignment. These limitations can be overcome by fuzzy methods, of which probabilistic methods form a subset. An innovative approach with applications to image search and image annotation is due to Barnard and Forsyth [2]. They propose to learn a generative hierarchical model with a fixed structure, i.e. the number of levels (l in Fig. 2) and the branching pattern are given. Each node of the hierarchy is associated with a probability distribution, $P(i|l, c)$, over terms and image features.

Each node specified by c and l generates images according to its distribution $P(i|l, c)$, and so therefore does every path that descends from the root to a leaf node. The parameters of the probability distributions for each node are learned from a training set, along with the probabilities of an image being generated by a particular path, and of an image feature being generated at a particular level of the path. This dual estimation is achieved using the expectation-maximisation algorithm on a set of training images. The fitted model admits to an interpretation in terms of a clustering by assigning each image to the path under which it has highest probability. It is these clusters that can subsequently be used for browsing. Although the clustering is hierarchical, the paper does not explore the possibility of hierarchical browsing, presumably because the number of clusters can be kept small enough to be displayed all at once. Nevertheless, the model stands out for its probabilistic integration of text and image data. The authors observe that images tend to improve the quality of the

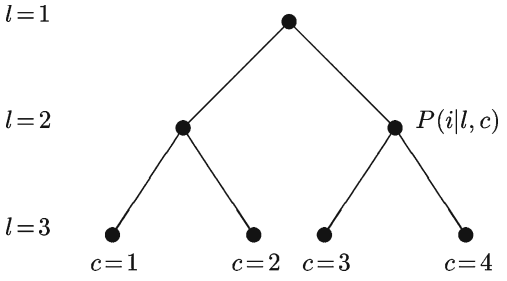

Fig. 2 Generative hierarchical model according to [2]. Given a path from root to a leaf node, each intermediate node generates terms and blobs depending on the level in the tree and the chosen path

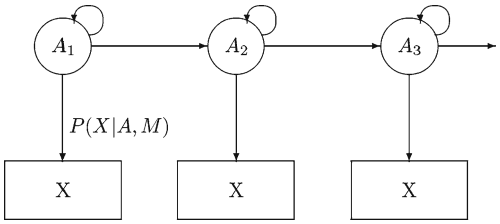

Fig. 3 A hidden markov model for organising photos into albums. A_i are hidden states (albums), X are image features and M represents model parameters

clustering (compared to clustering based on text only) as they provide information that is often complementary to textual descriptions.

A different probabilistic model had previously been proposed by [62] for the purpose of organising digital photographs into a set of clusters (or albums). Images are assumed to form an ordered set according to the time they were taken, so their generation can be modelled by a Hidden Markov Model (Fig. 3). The chain moves unidirectionally between states that represent individual albums A_i. Associated with each album is a probability density over image features X. The model only allows a flat clustering: Images are clustered into albums, but albums are not aggregated any further.

3.1.3 Summary

Hierarchies have the advantage that there is an abundance of relatively simple construction algorithms. Divisive, top-down k-means clustering runs in $O(n)$ but appears to give less coherent clusters than agglomerative clustering methods that typically run in $O(n^2)$.

If we consider the use of hierarchical structures not for representation but for interactive navigation, the labeling of intermediate clusters by cluster representatives becomes an important issue. If images are annotated either manually or through automated image annotation, there is a possibility of distinguishing cluster centroids on a textual basis (even though the selection of terms for intermediate clusters is far from trivial and an active research area, see for example [17]). If such annotation is not available, however, we have to resort to images as representations of cluster content. Although this is seldom acknowledged, cluster centroids rarely provide a meaningful summary of the content. In particular, it is difficult to represent the degree of generality of a cluster by virtue of an image as this typically admits to an interpretation at different semantic levels.

Psychologically, the process of navigating hierarchies is a double-edged sword: on the one hand browsing from the more general to the more specific can afford the impression of progressive refinement, on the other hand it may create a sense of lost opportunities if navigation is only along the vertical direction.

3.2 Static networks

The way knowledge is organised in the human brain is still poorly understood. The connectionist view holds that storage is not hierarchical but takes the form of associative structures in which concepts are linked together on the basis of semantic relationships [58, 64]. The ultimate physical realisation of nodes is in the form of

neurons or clusters of interconnected neurons. In these distributed structures the hierarchical nature of many data is implicit in the weights associated with pairs of connected nodes.

A popular computational model for retrieving information from these associative structures is that of automatic spreading activation [1]. The retrieval process starts by activating an initial set of nodes from which activation spreads towards connected nodes. Memory items that are directly or indirectly referred to from activated nodes will pass on activation to other associated nodes and so on. The retrieval process concludes when certain items accumulate enough activation to be retrieved.

Buckley and Salton [70] provide an overview of methods of spreading activation for information retrieval. Two broad approaches are discussed. In the first, activation is spread across networks of terms (obtained from term thesauruses) to achieve query expansion. In the second, the associative structures are the documents themselves linked by similarity. In both scenarioes, activation spreads automatically. The notion of interactive browsing of such networks is more recent and in image retrieval finds its earliest articulation in [19] (see below). The parallel spread of activation is here replaced by a serial user-driven search. As with hierarchical structures, the question how to establish links between objects is not obvious, whether the networks are used in an automated fashion or interactively.

Typically, networks are built on the basis of similarity data between objects. We discuss four different approaches that differ in the way edges are established between vertices.

3.2.1 Nearest neighbour networks

A significant work on interlinked information structures dates back to the mid-1980s [20]. The authors propose to structure a collection of documents as a network of nodes representing documents and terms with links between each type of node. Only links between a document and its most similar neighbour are considered, and similarly for terms. Although the structure is intended for automated search, the authors are aware that "as well as the probabilistic and cluster based searches, the network organisation could allow the user to follow any links in the network while searching for relevant documents. A special retrieval strategy, called browsing, could be based on this ability." (p. 380). However, the number of document-document edges does not exceed by much the number of documents, and the networks are disconnected rendering browsing along document-document nodes alone impractical.

The paper inspired subsequent work by Cox [18, 19]. Cox motivates associative structures for browsing by observing that "people remember objects by associating them with many other objects and events. A browsing system on a static database structure requires a rich vocabulary of interaction and associations." His idea is to establish a nearest neighbour network for each of a set of different object descriptors. As different features may be important to different users, Cox proposes to interconnect nearest neighbour networks to allow multi-modal browsing.

Unfortunately, Cox's work has not become very widely known. What may partly account for this is that, in the mid-1990s, CBIR had just found with QBE its first major research programme (as reflected in the review article of [33]). The methodological emphasis was on explicit query formulation and alternative approaches found themselves at the periphery of the research agenda.

3.2.2 Thresholded graphs

We encountered thresholded graphs in the previous section on hierarchical structures. Like nearest neighbour networks, thresholded graphs can be viewed as another attempt to present a salient subset of all the possible pair-wise relationships between images. Interestingly, they do not appear to have been used on their own either for visualisation or navigation purpose. The reason for this may be that the graphs resulting from thresholding are rather fragmented (hence their use in conjunction with clustering). This is illustrated in Fig. 4.

3.2.3 Pathfinder networks

The pathfinder algorithm [74, 75] was originally conceived for the analysis of similarity or proximity data obtained from psychological studies. An interesting property of such subjective similarity judgements is that they do not generally obey the triangle inequality, i.e. it is not generally true that $d(A, B) \leq d(A, C) + d(B, C)$ for all database items A, B, and C, where d is the dissimilarity between items. The motivation of the pathfinder algorithm is that links between items for which the triangle inequality does *not* hold are redundant and may be excluded from the graph. Formally, given the similarity between each of N items, the pathfinder algorithm removes an edge between vertices p and q if there exists a shorter path involving intermediate vertices. The length of the path, denoted D, is not the number of edges but a simple function of the weights associated with the set of edges E that constitute the path between p and q:

$$D(E) = \left(\sum_{e \in E} w_e^r \right)^{\frac{l}{r}}, \tag{1}$$

where r is a parameter that controls how individual edge weights contribute to the total weight. For high values, the total distance is largely determined by the edge with greatest weight. To ease the computational burden, paths that comprise more than a certain number of edges may be ignored. The minimum cost network thus obtained differs from minimum spanning trees and thresholded graphs to the extent that it takes into account the wider context of a link.

An early application of pathfinder networks is the work by Fowler et al. [28] which is concerned with representing co-citation relationships. Applications to image retrieval are more recent [13, 56]. In both works, pathfinder networks are constructed

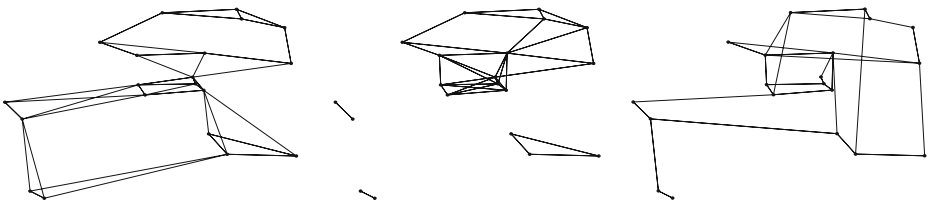

Fig. 4 Comparison of three different types of networks: a nearest neighbour graph, a thresholded graph and an NN^k network

from complete graphs in which any two images are connected by an edge that is weighed by their similarity. In [13], separate pathfinder networks are built for three different features (colour, layout and texture) to visually inspect the relative performance of these different features. In spite of the title of the paper, the networks are not used for browsing but for providing a global overview of the image collection.

Indeed, it appears that the principal application domain of pathfinder networks has so far been visual data mining ([11, 12]) not interactive browsing. The reason is quite likely to be found in the complexity, which is $O(n^2)$ (where n is the number of images) and thus prohibitive for collection sizes of practical significance. Moreover, visualisation and navigation place somewhat different structural demands on the networks. While visualisation requires the extraction of only the most salient structure, the objectives of browsing are better served by retaining some degree of redundancy.

3.2.4 NN^k Networks

NN^k networks were introduced as a novel browsing structure in [36] and were analysed more fully in [35]. They are based on a simple idea: instead of ranking images with respect to another image q according to their similarity under a fixed metric, the metric is parametrised in terms of feature-specific weights. For *each* parameter setting the image is recorded that is closest to q under that particular instantiation of the metric. This set of top-ranked images are referred to as the NN^k. NN stands for nearest neighbours and k denotes the number of features. The NN^k can be thought of as representing all the possible semantic facets of the focal image that lie within the representational scope of the feature set. Formally, p is the NN^k of q iff

$$\arg\min_i \left[\sum_{f=1}^{k} w_f d_f(i, q) \right] = p \tag{2}$$

for some parameter set $w = (w_1, w_2, \ldots, w_k)$ with $w_f \geq 0$, $\sum w_f = 1$ and feature-specific distance functions $d_f(\cdot, \cdot)$. w can be thought of as weights that determine the relative importance of individual features. Given a collection of images, the NN^k idea can be used to build image networks by placing an edge between image q and p if p is the NN^k of q.

The advantage of NN^k networks lies in their unbiased treatment of different visual features since each image is connected to all those images it is closest to under different feature weightings. This helps to alleviate the semantic bias that would result from imposing a fixed set of feature weights.

Structurally, NN^k networks resemble the hyperlinked network of the WWW, but they tend to exhibit a much better connectedness with only a negligible fraction of vertices not being reachable from the giant component. In collections comprising more than 100,000 images, the average number of links separating two images lies just above four [35].

Much like the WWW, NN^k networks can be navigated by following directed edges from one object to any of its neighbours. Since the exact position of a target is not known in advance, users must try to reach it based on local cues, that is the neighbours of the current node. Such local information supports a greedy search

process whereby users decide on the most favourable direction by selecting the image that comes closest to their target. That this method seems to work remarkably well in NN^k networks [38] might lie in the fact that at each step users can select from a varied set of neighbours. The process is similar to that of genetic algorithms. In the latter, a well-defined objective function exists which allows automating the selection step, meanwhile search in NN^k networks relies on user interaction. In both systems, variation at each step is crucial to be able to explore the space of possibilities effectively.

In [39], an NN^k networks was constructed for more than 1 million images from the WWW. Although network construction is quadratic in the number of images and for such large collections may take several days on one processor, interaction during search is in real time. The currently selected image is displayed in the centre, its NN^k are positioned according to the number of different feature combinations for which they are ranked closest to the central image.

To help appreciate the structural differences between NN^k networks, threshold graphs and nearest neighbour networks, we have applied each of the three different methods to a synthetic two-dimensional dataset (Fig. 4). To make the comparison more meaningful, we kept the average number of neighbours per node the same across all networks. This can be achieved by simply fixing the number of neighbours for the NN^k network and the nearest network (here three), and choosing a suitable cutoff weight for thresholding the graph to yield the same average across all nodes. Because the thresholded graph takes into account distances, not ranks, the connections are more localised than for the other two networks (resulting in remote nodes being isolated from the rest). The NN^k network lacks some of the short range connections of the nearest neighbour network but offers a number of long-range connections instead. These account for the small-world properties and the good overall connectedness of the structure.

3.2.5 Navigable hypercubes

Similarly inspired but different in detail from NN^k networks is the idea of navigable hypercubes proposed in [51]. The paper is concerned with objects that can be represented as binary vectors $[0, 1]^k$ where a 1 indicates that the corresponding feature is present. Each of the 2^k possible vectors forms a node in the graph and two nodes are connected if they differ in exactly one component. The objects are uniquely assigned to nodes according to their feature representations and users browse the hypercube by selecting and deselecting features. While NN^k networks link individual objects according to the similarity with respect to different features, the hypercube model links *groups* of objects (united by the same feature representation) if they differ only with respect to one feature. The hypercube structure is useful when features are meaningful to the user such as “published in 2002” or “conference article” but much less so for the kind of visual, continuous features typically employed for image retrieval.

3.2.6 Summary

Networks have the advantage over hierarchies that they allow navigation to be less constrained. At the same time, it is more difficult to provide a global overview of

the content. It becomes increasingly important to organise objects in the network such that the local neighbourhood of the currently selected object contains sufficient information for users to decide where to go next.

It is instructive to consider the question why humans rarely employ networks for organising objects in the real world. Partly this is because hierarchies can be indexed and therefore accessed easily. Ultimately, it may have to do with the fact that hierarchies form planar graphs (graphs that can be drawn in the plane such that no two edges intersect) while all but the simplest networks do not. Hierarchies thus have a practical advantage: children of a node can always be arranged such that they are adjacent (e.g. books in a library organised according to a conceptual hierarchy). This is not generally possible for networks in the real world because objects would have to be present in multiple copies and space would need to be abundant. Importantly, both limitations disappear in virtual environments where the networks are therefore of much greater practical interest.

A crucial question pertaining to precomputed structures is how to weigh different features when precomputing similarities. NN^k networks attempt to solve this problem by establishing nearest neighbour links under all possible instantiations of a parametrised metric. Although the structure is rigid, the networks can be viewed as representing the set of all the similarity relations different users may see within an image set.

3.3 Dynamic structures

In this section, we turn to hybrid models that have grown out of the QBE tradition. Many seek to display a much larger set of search results than conventional query based systems thus catering for users with only poorly defined information needs. Some form of feedback is typically allowed to let users home in on subsets of the search results and queries are often formulated much less explicitly.

3.3.1 Ostensive browsing

The ostensive model proposed by [9] is iterated QBE in disguise but the query only emerges through the interaction of the user with the collection. The impression for the user is that of navigating along a dynamically unfolding tree structure. While originally developed for textual retrieval of annotated images, the ostensive model is equally applicable to visual features ([8, 84]).

Relevance feedback takes the form of selecting an image from those displayed. A new query is formed as the weighted sum of the features of this and previously selected images. For example, in [84] image content is described in the form of a colour histogram c. Given a sequence of selected images, the colour representation of the new query is given by $\sum w_i c_i$ with $w_i = 2^{-i}$, where images are indexed starting with the most recently selected image.

The display model is that of a dynamically growing tree structure: images closest under the current query are displayed in a fan-like pattern to one side of the currently selected image. Users can select an image from the retrieved set which is placed in the centre and a new set of images are retrieved in response. Since previous images are

kept on the display the visual impression for the user is that of establishing a browsing path through the collection. In [84] the browsing path is displayed in a Fisheye view as pioneered in [30].

The ostensive model attempts to track changing information needs by continuously updating the query. Which set of images are retrieved depends on which path the user has travelled to arrive at the current image. Because the number of such different paths grows quickly with the size of the image collection, it is impractical to compute a global structure beforehand. Nonetheless, for the user the impression is one of navigating in a relatively unconstrained manner through the image space. Unlike many other relevance feedback systems, users do not have to rank or label images, or change their relative location. The interaction is thus light and effective.

3.3.2 Dimensionality reduction

The model of [9] is not concerned with the question how to optimally display the set of retrieved images. In fact, given that much of the space is taken up by the browsing path, the size of the retrieved set is kept relatively small (six in [84]) and correspondingly small is the scope and the need for layout optimisation. For larger solution sets, users would benefit from a more structured view of the images. Dimensionality reduction methods try to achieve this.

A popular way of organising small collections is multidimensional scaling or MDS (e.g. [63, 68] and more recently [50] and [92]). MDS is a family of techniques that solve a nonlinear optimisation problem by determining a mapping into two or three dimensions that best approximates all the high-dimensional pairwise distances between data points.

The quadratic complexity of MDS makes it less attractive as a means of organising large image sets at run time. Indeed, scaling is most often applied to a smaller set of search results ([52, 69, 72, 73]). In [69], the computation time is given with four seconds for 100 images in 1998, which seems quite acceptable.

The work of Rubner et al. [69] is important on several accounts. In addition to motivating the use of signatures over histograms and describing the Earth mover's distance as an appropriate metric for signatures, it applies MDS to the problem of iterated image search. The paper advances beyond its otherwise very similar precursor [68] by emphasising the use of MDS for iterative organisation of query results (rather than for visualising the entire collection). The search process is one of continuous refinement from an initially large pool of images to an increasingly narrow set. This is accomplished by users selecting images from the MDS display as their query in addition to specifying the number m of images to be displayed in return. Each query entails the computation of $\binom{m}{2}$ distances. Santini and Jain [72] and [73] differ from [69] in that users are allowed to provide relevance feedback by forming image groupings and enabling the system to learn from the new configuration.

A number of works (e.g. [41, 44, 57] and [55]) achieve dimensionality reduction of the retrieved set through conventional principal component analysis (PCA), which is a linear projection technique. The advantage of PCA is computational efficiency but, unlike general MDS, it requires images to be representable in a vector space and only allows linear transformations.

The authors of [55] propose a relevance feedback mechanism similar to [72] and [73] whereby feature weights are estimated based on image groups composed by the

user. PCA is then applied again to the newly scaled feature space. However, none of the above works consider imposing additional structure upon the retrieved set.

The authors of [61] recognise the limitations of dimensionality reduction for displaying large image sets. In their work a Sammon mapping [71] is applied to the entire image set after first achieving dimensionality reduction by PCA, thus speeding up the final mapping. The Sammon mapping is a particular multi-dimensional scaling method that aims to preserve the local topology of the dataset by placing greater emphasis on the preservation of small inter-point distances. In addition to the Sammon projection, images are clustered in the original feature space. The positions of the cluster centroids are determined by the Sammon projection. The two organising principles of dimensionality reduction and hierarchical clustering thus coexist independently. As in [69] users have the impression of homing in on a small set of images. Unlike in [69], however, the layout is not recomputed at each step, but relies on the initial, global mapping. This arguably comes at the price of reduced visual coherence but allows navigation to be fast. The direction is from top to bottom and arguably less suited for exploring a database than [69] where users can navigate into different areas of the image by keeping m large.

In [59], a number of non-linear projection methods are explored, most notably isometric mapping (ISOMAP) [82], stochastic neighbour embedding (SNE) [40], local linear embedding (LLE) [67] and combinations between the first and the last two (ISOSNE and ISOLLE). Each of these methods aim to preserve more of the local structure than general MDS techniques. ISOMAP, for example, involves building a neighbourhood graph using the k closest points in the original high-dimensional space. The distance between any two points is the length of the shortest path on the resulting graph. MDS is subsequently applied to these distances to achieve dimensionality reduction. The interaction model of [59] is an iterative process that begins by displaying an overview of the collection in terms of a few representative images obtained by clustering the images in the projected space (using k-means). Choosing one of the cluster centroids retrieves the images closest to it. The authors conclude that for a number of different tasks including search, the SNE and ISOSNE methods seem to capture the structure of the collection best but also take substantially longer to run than LLE and ISOLLE.

3.3.3 Clustering search results

A dynamic method that allows users to home in on a small target set of images by combining clustering with QBE is developed in [85] under the term filter image browsing. Users choose from the displayed images, which are meant to be representative of the "active set" that initially comprises all images. The system then ranks all images from the current active set according to their similarities to the chosen images. A fixed percentage is kept and forms the active set for the next step. The ambition of filter image browsing is to combine the adaptivity of QBE with the structural richness of hierarchical browsing. It is noteworthy, however, that the method is less suited for freely roaming an image collection for once an image has been excluded from the active image set, it will remain unreachable in all subsequent iterations. The possibility of altering the distance metric under relevance feedback is mentioned but not investigated in practice.

Fig. 5 *First row*: a query consisting of a Boolean composition of regions. *Second row*: matching images. (Courtesy of J Fauqueur)

3.3.4 Region composition

The recent study of [25] is motivated by the observation that users may often not have the right image to query a system with, and that submitting hand-drawn sketches has limited expressivity and may not appeal to many users. The proposed solution lets users express their mental search image by composing elementary regions, that by themselves may carry little semantics but do in combination. The regions are obtained through unsupervised clustering of the color descriptors of image segments from a training set, and together they form what the authors call a "region photometric thesaurus". Regions can be combined in Boolean expressions such as shown in Fig. 5. Images are retrieved if their regions satisfy the compositional query.

3.3.5 Summary

The potential disadvantage of static networks or hierarchies is the limited scope for learning from the user during the interaction. The models discussed in the previous section require a variable amount of computation during the interaction (e.g. by performing a nearest neighbour search or clustering the search results) but have the ability to adapt to feedback from the user as can be seen most strikingly in Campbell's ostensive browsing model and the filter image browsing model. The question how these systems scale to several millions of images has not yet been investigated experimentally but it is likely that much algorithmic optimisation and hardware support is needed in order to apply the techniques successfully to real-world, large-scale applications.

4 Discussion

A comparison of most of the methods discussed in Section 3 is given in Table 1. We have organised the methods according to seven criteria which identify the columns of the table: 'Dynamic' associates the model with either Sections 3.1 and 3.2 on static structures or Section 3.3 on dynamic models. 'Topology' refers to whether the structure is a network or a hierarchy. If the model displays images following dimensionality reduction, we refer to the resulting topology as a map (M in the table). 'Generative method' refers to the technique used to build the structure (C = clustering, DR = dimensionality reduction). 'Open ended' assesses whether users can continue searching and possibly change their information need without having

Table 1 Different browsing models and their properties. See text for explanation of criteria

Author	Dynamic	Topology	Generative method	Open ended	O(offline)	O(online)	Collection size
[95]	–	H	AC	–	n^2	1	4,000
[14, 15]	–	H	AC	✓	n^2	1	10,000
[46]	–	H	AC	✓	n^2	1	3,856
[61]	–	H	AC	✓	n^2	1	6,100
[91]	–	H	SOM	✓	n^2	1	2,000
[2]	–	H	PR	–	n^2	1	3,000
[62]	–	H	PR	–	n^2	1	1,298
[32]	–	H	IB	–	n^2	1	?
[16]	–	H	DC	–	n^2	m	46,331
[61]	–	H	DC	✓	n^2	1	6,100
[13]	–	N	PF	✓	n^4	1	279
[18, 19]	–	N	NN	✓	n^2	1	< 100
[13]	–	N	PF	✓	n^4	1	279
[56]	–	N	PF	✓	n^4	1	75
[51]	–	N	O	✓	n^2	1	7,541
[35]	–	N	NN^k	✓	n^2	1	32,318
[68]	–	M	MDS	✓	n^2	1	500
[69]	–	M	MDS	✓	1	nm^2	1,000
[44]	–	M	PCA	–	1	m^2	3,500
[41]	–	M	PCA	–	n^2	m^2	10,000
[57]	–	M	PCA	✓	1	m^2	9,807
[59]	–	M	ISOLLE	–	n^2	1	10,000
[85]	✓	H	O	–	1	n	10,000
[84]	✓	H	NN	✓	1	n	800
[61]	✓	M	Sammon	✓	n^2	1	6,100
[72, 73]	✓	M	MDS	–	1	nm^2	3,000
[55]	✓	M	PCA	–	1	m^2	142

H = hierarchies, N = networks, M = maps, AC = agglomerative clustering, DC = divisive clustering, SOM = self-organising map, PR = probabilistic model, PCA = principal component analysis, MDS = multi-dimensional scaling, PF = pathfinder algorithm, NN = nearest neighbour, IB = information bottleneck principle, O = other

to start from the beginning. 'O(offline)' and 'O(online)' gives the complexity of building the structure and for interacting with it at runtime (we use n for the size of the collection and m for the size of the retrieved set).

As has been emphasised throughout the paper, a key challenge pertaining to precomputed structures as discussed in Sections 3.1 and 3.2 is that by being precomputed users are not generally in a position to remould the structure according to their own preferences. This seems necessary, however, as the structures are almost always constructed by fixing the distance metric and applying that same metric across the entire collection. The advantage of fast navigation comes at the price of users being no longer able to impose their own perception of similarity. Precomputed structures seem therefore to deride the principal tenet that motivates relevance feedback techniques. As observed in [100] in the context of hierarchical structures, "the rationale of relevance feedback contradicts that of pre-clustering." However, relevance feedback techniques tend to scale badly to very large collections and often result in relatively fast convergence to a small set of images. As seen in some of the

dynamic systems of Section 3.3, the scope for free exploration is often more limited than in precomputed structures. It remains a challenge to strike the right balance between the two opposing demands of real-time interaction and the ability to support the search through feedback loops.

Given the range of different approaches and the fact that some solve certain problems better than others, it is clearly conceivable to merge different models and offer users a selection of topologies and interaction methods for the same collection.

In addition to those already mentioned, there are a number of other exciting and important issues, a solution to which should lead to a new generation of smarter, more versatile browsing models. Some of these are sketched below.

Scalable update Shaping large collections into a browsable structure can be computationally costly as one typically has to compute all pairwise distances between images. This seems acceptable if the effort needs to be expended only once but many collections are dynamic with new images regularly being added and others removed. An update should not involve a complete recomputation of the structure. The extent to which the above models lend themselves to an efficient update is seldom investigated (see [16] for an exception).

Fluid structures The systems we have discussed either involve a precomputed structure or initiate a new query at every step. Systems of the first kind are often too rigid, systems of the second too slow for large collections. What may hold promise are hybrid structures that are partially precomputed but flexible enough to remain responsive to relevance feedback. In the early work of Minka and Picard [54] it is suggested to precompute a large number of image groupings by applying a variety of different similarity criteria, or 'society of models'. Their method seeks to capture as many of the potentially relevant groupings beforehand. Given a set of positive images, groupings are retrieved that exhibit the greatest overlap with the selected images. Though less suited for exploration, this approach gives an idea of the kind of semi-fluid interaction that one may wish to achieve in future browsing models.

Long-term learning By searching interactively for images users continuously provide implicit relevance feedback. In addition to exploiting this information for the current search session, it would be desirable to equip systems with some form of long-term memory that aggregates the entire history of their interactions with users *and* provide means to utilise this information. Considering the proliferation of large-scale systems on the WWW (e.g. Flickr) and a growing community of active users, this might no longer be a significant practical problem. A related challenge that has not yet been explored is to build into browsing structures knowledge about the perceptual similarities of real users that might have been obtained from other sources. The system in [93] is a recent example of an initiative that attempts to gather such data, and subsequently use it to guide the construction of more relevant browsing structures.

5 Conclusions

Throughout the last decade, the dominant paradigm in the field of CBIR revolved around the idea that users supplied an explicit query to a retrieval system. This paper

is motivated by our view that some of the problems associated with this tradition may be better addressed within a more interactive browsing framework. It is noteworthy that the different models we discussed are not only numerous but also very diverse. This testifies to the fact that browsing also has its challenges and trade-offs and that while a model may solve one problem, none solves them all. By having brought into clearer focus the virtues and challenges of browsing methods, we hope to inspire more concerted future research in the area.

References

1. Anderson J (1983) A spreading activation theory of memory. J Verbal Learn Verbal Behav 22:261–295
2. Barnard K, Forsyth D (2001) Learning the semantics of words and pictures. In: Proc IEEE int'l conf computer vision, vol 2. IEEE, Piscataway, pp 408–415
3. Beckmann N, Kriegel H-P, Schneider R, Seeger B (1990) The R*-tree: an efficient and robust access method for points and rectangles. In: Proc int'l conf management of data, Atlantic City, 23–26 May 1990, pp 322–331
4. Bentley J (1975) Multidimensional binary search trees used for associative searching. Commun ACM 18(9):509–517
5. Berkhin P (2002) Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA
6. Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) When is 'nearest neighbour' meaningful? In: Proc int'l conf data theory, Jerusalem, 10–12 January 1999, pp 217–235
7. Boucheron L, Creusere C (2005) Lossless wavelet-based compression of digital elevation maps for fast and efficient search and retrieval. IEEE Trans Geosci Remote Sens 43(5):1210–1214
8. Browne P, Smeaton A (2004) Video information retrieval using objects and ostensive relevance feedback. In: ACM symp applied computing. ACM, New York, pp 1084–1090
9. Campbell I (2000) The ostensive model of developing information-needs. PhD thesis, University of Glasgow
10. Carmel E, Crawford S, Chen H (1992) Browsing in hypertext: a cognitive study. IEEE Trans Syst Man Cybern 22:865–884
11. Chen C, Kuljis J (2003) The rising landscape: a visual exploration of superstring revolutions in physics. J Am Soc Inf Sci Technol 54(5):435–446
12. Chen C, Morris S (2003) Visualizing evolving networks: minimum spanning trees versus pathfinder networks. In: IEEE symp information visualization. IEEE, Piscataway, pp 67–74
13. Chen C, Gagaudakis G, Rosin P (2000) Similarity-based image browsing. In: Proc int'l conf intelligent information processing, Beijing, 22 August 2000, pp 206–213
14. Chen J, Bouman C, Dalton J (1998) Similarity pyramids for browsing and organization of large image databases. In: Proc SPIE conf human vision and electronic imaging III, vol 3299. SPIE, Bellingham, pp 563–575
15. Chen J, Bouman C, Dalton J (2000) Hierarchical browsing and search of large image databases. IEEE Trans Image Process 9(3):442–455
16. Cheung S, Zakhor A (2005) Fast similarity search and clustering of video sequences on the World-Wide-Web. IEEE Trans Multimedia 7(3):524–537
17. Clough P, Joho H, Sanderson M (2005) Automatically organising images using concept hierarchies. In: Proc ACM multimedia workshop (SIGIR), Singapore, 6–11 November 2005
18. Cox K (1992) Information retrieval by browsing. In: Proc int'l conf new information technology, Hong Kong, 30 November–2 December 1992
19. Cox K (1995) Searching through browsing. PhD thesis, University of Canberra
20. Croft B, Parenty T (1985) Comparison of a network structure and a database system used for document retrieval. Inf Syst 10:377–390
21. Crucianu M, Ferecatu M, Boujemaa N (2004) Relevance feedback for image retrieval: a short review. In: State of the art in audiovisual content-based retrieval, information universal access and interaction including datamodels and languages (DELOS2 report)

22. Datta R, Joshi D, Li J, Wang J (2008) Image retrieval: ideas, influences and trends of the new age. ACM Trans Comput Surv (in press)
23. Descampe A, Vleeschouwer C, Iregui M, Macq N, Marqués F (2007) Prefetching and caching strategies for remote and interactive browsing of JPEG2000 images. IEEE Trans Image Process 16(5):1339–1354
24. Duda R, Hart P, Stork D (2001) Pattern recognition. Wiley, New York
25. Fauqueur J, Boujemaa N (2006) Mental image search by boolean composition of region categories. Multimed Tools Appl 31(1):95–117
26. Feng S, Manmatha R, Lavrenko V (2004) Multiple Bernoulli relevance models for image and video annotation. In: Proc int'l conf computer vision and pattern recognition. IEEE, Piscataway, pp 1002–1009
27. Forsyth D (2001) Benchmarks for storage and retrieval in multimedia databases. In: Proc SPIE conf storage and retrieval for media databases, vol 4676. SPIE, Bellingham, pp 240–247
28. Fowler R, Wilson B, Fowler W (1992) Information navigator: an information system using associative networks for display and retrieval. Technical report, Department of Computer Science, University of Texas, No. 92-1
29. Fukunaga K, Narendra P (1975) A branch and bound algorithm for computing *k*-nearest neighbors. IEEE Trans Comput 24(7):750–753
30. Furnas G (1986) Generalized fisheye views. In: Proc SIGCHI conf human factors in computing systems, Boston, 13–17 April 1986, pp 16–23
31. Gevers T, Smeulders A (2004) Content-based image retrieval: an overview. In: Medioni G, Kang S (eds) Emerging topics in computer vision. Prentice Hall, Englewood Cliffs
32. Goldberger J, Gordon S, Greenspan H (2006) Unsupervised image set clustering using an information theoretic framework. IEEE Trans Image Process 15(2):449–458
33. Gupta A, Jain R (1997) Visual information retrieval. Commun ACM 40(5):71–79
34. Guttmann A (1984) R-trees: a dynamic index structure for spatial searching. In: Proc ACM int'l conf management of data (SIGMOD), ACM, New York, pp 47–57
35. Heesch D (2005) The NN^k technique for image searching and browsing. PhD thesis, Imperial College London
36. Heesch D, Rüger S (2004) NN^k networks for content-based image retrieval. In: Proc European conf information retrieval, LNCS 2997. Springer, Berlin Heidelberg New York, pp 253–266
37. Heesch D, Rüger S (2006) Interaction models and relevance feedback in content-based image retrieval. In: Zhang Y-J (ed) Semantic-based visual information retrieval. Idea-Group, Harrisburg, pp 160–186
38. Heesch D, Pickering M, Yavlinsky A, Rüger S (2004) Video retrieval within a browsing framework using keyframes. In: Proc TREC video. NIST, Gaithersburg
39. Heesch D, Yavlinsky A, Rüger S (2006) NN^k networks and automated annotation for browsing large image collections from the World Wide Web. In: Proc ACM int'l conf multimedia (SIGMM). ACM, New York, pp 220–224
40. Hinton G, Roweis S (2002) Stochastic neighbour embedding. In: Advances in neural information processing systems, vol 15. MIT, Cambridge, pp 833–840
41. Hiroike T, Musha Y, Sugimoto A, Mori Y (1999) Visualization of information spaces to retrieve and browse image data. In: Visual information systems. Morgan Kaufmann, San Francisco, pp 155–162
42. Jacobs C, Finkelstein A, Salesin D (1995) Fast multiresolution image querying. Technical report, University of Washington, US
43. Katayama N, Satoh S (1997) SR-tree: an index structure for high-dimensional nearest neighbour queries. In: Proc ACM int'l conf management of data (SIGMOD), ACM, New York, pp 369–380
44. Keller I, Meiers T, Ellerbrock T, Sikora T (2001) Image browsing with PCA-assisted user-interaction. In: IEEE workshop content-based access of image and video libraries. IEEE, Piscataway, pp 102–108
45. Kohonen T (2001) Self-organizing maps, volume 30 of Springer series in information sciences. Springer, Berlin Heidelberg New York
46. Krishnamachari S, Abdel-Mottaleb M (1999) Image browsing using hierarchical clustering. In: IEEE symp computers and communications. IEEE, Piscataway, pp 301–307
47. Kurniawati R, Jin J, Shepherd J (1997) Techniques for supporting efficient content-based retrieval in multimedia databases. Aust Comput J 29(4):122–130

48. Laaksonen J, Oja E, Koskela M, Brandt S (2000) Analyzing low-level visual features using content-based image retrieval. In: Proc int'l conf neural information processing, Taejon, 14–18 November 2000
49. Lavrenko V, Manmatha R, Jeon J (2003) A model for learning the semantics of pictures. In: Advances in neural information processing systems, vol 16. MIT, Cambridge
50. Lim S, Chen L, Lu G, Smith R (2005) Browsing texture image databases. In: Proc int'l conf multimedia modelling. IEEE, Piscataway, pp 328–333
51. Liu T, Joung Y (2004) Multi-dimension browse. In: Proc IEEE int'l conf computer software and applications. IEEE, Piscataway, pp 480–485
52. MacCuish J, McPherson A, Barros J, Kelly P (1996) Interactive layout mechanisms for image database retrieval. In: Proc SPIE conf visual data exploration and analysis III, vol 2656. SPIE, Bellingham, pp 104–115
53. Milanese R, Squire D, Pun T (1996) Correspondence analysis and hierarchical indexing for content-based image retrieval. In: Proc IEEE int'l conf image processing. IEEE, Piscataway, pp 859–862
54. Minka T, Picard R (1996) Interactive learning using a society of models. In: Proc IEEE conf computer vision and pattern recognition. IEEE, Piscataway, pp 447–452
55. Moghaddam B, Tian Q, Lesh N, Shen C, Huang T (2004) Visualization and user-modeling for browsing personal photo libraries. Int J Comput Vis 56(1–2):109–130
56. Mukhopadhyay R, Ma A, Sethi I (2004) Pathfinder networks for content based image retrieval based on automated shape feature discovery. In: Proc IEEE int'l symp multimedia software engineering. IEEE, Piscataway, pp 522–528
57. Musha Y, Hiroike A, Mori Y, Sugimoto A (1998) An interface for visualizing feature space in image retrieval. In: Proc IAPR workshop machine vision applications, Chiba, 17–19 November 1998, pp 447–450
58. Newell A (1990) Unified theories of cognition. Harvard University Press, Cambridge
59. Nguyen G, Worring M (2008) Interactive access to large image collections using similarity-based visualization. J Vis Lang Comput (in press)
60. Obdržálek S, Matas J (2005) Sub-linear indexing for large scale object recognition. In: Proc conf British machine vision, Versailles, 5–8 September 2005, pp 1–10
61. Pečenović Z, Do M, Vetterli M, Pu P (2000) Integrated browsing and searching of large image collections. In: Proc int'l conf advances in visual information systems, LNCS 1929. Springer, Berlin Heidelberg New York, pp 279–289
62. Platt J, Czerwinski M, Field B (2002) PhotoTOC: automatic clustering for browsing personal photographs. Technical report, Microsoft Research
63. Rodden K, Basalaj W, Sinclair D, Wood K (2001) Does organization by similarity assist image browsing? In: Proc int'l conf computer human interaction, New Orleans, 5–10 August 2001, pp 190–197
64. Rogers T, McClelland J (2006) Semantic cognition: a parallel distributed processing approach. MIT, Cambridge
65. Roussinov D, Chen H (1998) A scalable self-organizing map algorithm for textual classification: a neural network approach to thesaurus generation. Commun Cogn 15(1–2):81–112
66. Roussopoulos N, Kelley S, Vincent F (1995) Nearest neighbor queries. In: Proc ACM int'l conf management of data (SIGMOD). ACM, New York
67. Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
68. Rubner Y, Guibas L, Tomasi C (1997) The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In: Proc ARPA image understanding workshop, New Orleans, May 1997, pp 661–668
69. Rubner Y, Tomasi C, Guibas L (1998) A metric for distributions with applications to image databases. In: Proc IEEE int'l conf computer vision. IEEE, Piscataway, pp 59–66
70. Salton G, Buckley C (1988) On the use of spreading activation methods in automatic information. In: Proc ACM int'l conf information retrieval (SIGIR). ACM, New York, pp 147–160
71. Sammon J (1969) A nonlinear mapping for data structure analysis. IEEE Trans Comput C-18(5):401–409
72. Santini S, Jain R (2000) Integrated browsing and querying for image databases. IEEE Multimed Mag 7(3):26–39
73. Santini S, Gupta A, Jain R (2001) Emergent semantics through interaction in image databases. IEEE Trans Knowl Data Eng 13(3):337–351

74. Schvanefeldt R (1990) Pathfinder associative networks: studies in knowledge organization. In: Ablex series in computational sciences. Ablex, Norwood
75. Schvaneveldt R, Durso F, Dearholt D (1989) Network structures in proximity data. In: Bower GH (ed) The psychology of learning and motivation. Academic, London, pp 249–284
76. Sclaroff S, Taycher L, La Cascia M (1997) ImageRover: a content-based image browser for the World Wide Web. Technical report, Boston University
77. Sloutsky V (2003) The role of similarity in the development of categorization. Trends Cogn Sci 7(6):246–252
78. Smeulders A, Worring M, Santini S, Gupta A, Jain R (2000) Content based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380
79. Smith J, Chang S-F (1996) VisualSEEk: a fully automated content-based image query system. In: Proc ACM int'l conf multimedia (SIGMM). ACM, New York
80. Spence R (1999) A framework for navigation. Int J Hum Comput Stud 51:919–945
81. Tenenbaum J (2006) Theory-based bayesian models of inductive learning and reasoning. Trends Cogn Sci 10(7):309–317
82. Tenenbaum J, de Silva V, Langford J (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
83. Tishby N, Pereira F, Bialek W (1999) The information bottleneck method. In: Proc allerton conf communication, control and computing, Monticello, September 1999, pp 368–377
84. Urban J, Jose J, van Rijsbergen C (2003) An adaptive approach towards content-based image retrieval. In: Proc int'l workshop content-based multimedia indexing, Rennes, 22–24 September, pp 119–126
85. Vendrig J, Worring M, Smeulders A (1999) Filter image browsing: exploiting interaction in image retrieval. In: Visual information and information systems, Amsterdam, 2–4 June 1999, pp 147–154
86. Wang Q, You S (2006) Fast similarity search for high-dimensional datasets. In: Proc IEEE int'l symp multimedia. IEEE, Piscataway
87. Weber R, Blott S (1997) An approximation based data structure for similarity search. Technical Report 24, ETH Zurich, Switzerland
88. Weber R, Schek J-J, Blott S (1998) A quantitative analysis and performance study for similarity-search methods in high-dimensional space. In: Proc int'l conf very large databases, New York, 24–27 August 1998, pp 194–205
89. White D, Jain R (1996) Similarity indexing with the SS-tree. In: Proc IEEE int'l conf data engineering. IEEE, Piscataway, pp 516–523
90. Wolfram S (2004) A new kind of science. Wolfram, Champaign
91. Yang C (2004) Content-based image retrieval: a comparison between query by example and image browsing map approaches. J Inf Sci 30(3):254–267
92. Yang J, Fan J, Hubball D, Gao Y, Luo H, Ribarsky W, Ward M (2006) Semantic image browser: bridging information visualization with automated intelligent image analysis. In: IEEE symp visual analytics science and technology. IEEE, Piscataway, pp 191–198
93. Yavlinsky A, Heesch D (2007) An online system for gathering image similarity judgements. In: Proc ACM int'l conf multimedia (SIGMM). ACM, New York, pp 565–568
94. Yavlinsky A, Schofield E, Rüger S (2005) Automated image annotation using global features and robust nonparametric density estimation. In: Proc int'l conf video and image retrieval, LNCS 3568. Springer, Berlin Heidelberg New York, pp 507–517
95. Yeung M, Liu B (1995) Efficient matching and clustering of video shots. In: Proc IEEE int'l conf image processing. IEEE, Piscataway, pp 338–341
96. Yeung M, Yeo B-L (1997) Video visualization for compact presentation and fast browsing of pictorial content. IEEE Trans Circuits Syst Video Technol 7(5):771–785
97. Zass R, Shashua A (2005) A unified treatment of hard and probabilistic clustering methods. In: Proc int'l conf computer vision. IEEE, Piscataway
98. Zhang H, Zhong D (1995) A scheme for visual feature based image indexing. In: Proc SPIE/IS&T conf storage and retrieval for image and video databases III, vol 2420. SPIE, Bellingham, pp 36–46
99. Zhang R, Zhang Z, Li M, Ma W-Y, Zhang H-J (2005) A probabilistic semantic model for image annotation and multi-modal image retrieval. In: Proc int'l conf computer vision. IEEE, Piscataway, pp 846–851
100. Zhou X, Huang T (2003) Relevance feedback in image retrieval: a comprehensive review. ACM Multimed Syst 8(6):536–544

Daniel Heesch obtained his PhD in computer science in 2005 from Imperial College London for his study on network structures for interactive image retrieval. He previously obtained degrees in mathematics (2005) and biology (2000) from Open University and Oxford University, respectively. He is currently a postdoctoral researcher in the Communications and Signal Processing Group at Imperial College London where he works on models for contextual image interpretation.