# Visualisation and Browsing of Image Databases

William Plant[1] and Gerald Schaefer[2]

[1] School of Engineering and Applied Science
Aston University
Birmingham, U.K.
plantwr1@aston.ac.uk

[2] Department of Computer Science
Loughborough University
Loughborough, U.K.
gerald.schaefer@ieee.org

In this chapter we provide a comprehensive overview of the emerging field of visualising and browsing image databases. We start with a brief introduction to content-based image retrieval and the traditional query-by-example search paradigm that many retrieval systems employ. We specify the problems associated with this type of interface, such as users not being able to formulate a query due to not having a target image or concept in mind. The idea of browsing systems is then introduced as a means to combat these issues, harnessing the cognitive power of the human mind in order to speed up image retrieval. We detail common methods in which the often high-dimensional feature data extracted from images can be used to visualise image databases in an intuitive way. Systems using dimensionality reduction techniques, such as multi-dimensional scaling, are reviewed along with those that cluster images using either divisive or agglomerative techniques as well as graph-based visualisations. While visualisation of an image collection is useful for providing an overview of the contained images, it forms only part of an image database navigation system. We therefore also present various methods provided by these systems to allow for interactive browsing of these datasets. A further area we explore are user studies of systems and visualisations where we look at the different evaluations undertaken in order to test usability and compare systems, and highlight the key findings from these studies. We conclude the chapter with several recommendations for future work in this area.

## 1 Introduction

Nowadays, the majority of people possess some form of digital camera to use in their everyday lives. Devices range from relatively low quality web cameras, to medium range cameras integrated into mobile devices, to higher quality cameras aimed at the average user, on to high-end cameras used

W. Lin et al. (Eds.): Multimedia Analysis, Processing & Communications, SCI 346, pp. 3–57.
springerlink.com 

by professional photographers. Affordability of devices and storage media coupled with increased capabilities and the 'to hand' availability of camera equipment has led to a dramatic increase in the number of digital images the average end user creates and stores.

With the reduction in digital photography costs, a shift in the attitude towards photo taking can be observed. Users tend to take more images now than before, particularly of the same objects or scene (e.g. from different perspectives) [25]. This is certainly a change from the past, where one would generally be concerned about the number of exposures left on the current film roll or the cost of developing photographs, whereas a digital camera user not happy with a photo can simply delete it from the camera's memory and images can be printed on home printers.

Personal image collections nowadays are typically in the range of hundreds to thousands of images. The rapid increase in the number of digital images taken by individuals has also caused an exponential growth in the number of images available online. Social networking sites allow users to instantly share images with friends, family or a wider community of users that also have the ability to comment and even 'tag' who or what may be in an image.

Commercially, professional photography companies may store millions of digital images in their databases [50]. These are generally manually annotated image collections used by journalists from a variety of publications to search for images suited to their particular needs. As one can imagine, the search for any particular image in collections of either personal or commercial magnitude can be tiresome and exhaustive. Generally, images are arranged in a one-dimensional linear arrangement, whereby an image has no correlation to any of its neighbours. Images are usually grouped together in a manually named folder or on the basis that they were uploaded to the computer at the same time.

This organisation of images is not ideal for a variety of reasons. Firstly, the cost of storage media has dramatically decreased whilst storage capacity has increased. Therefore an average end user may take many photos of many different events (such as birthdays, holidays etc.) on a camera before uploading them to their computer. If not sorted manually, multiple events may get grouped together, potentially making it difficult for the user to locate specific images in the future.

This leads to a second issue of manually annotating folders. If images of multiple events are stored in the same folder, it is difficult to describe the ambiguity of the content contained within it using just a folder name. Typically the date of the camera upload will be chosen, but this could become rather meaningless after a long period of time. Rodden and Wood [62] demonstrated in their analysis of digital photograph management that users are generally unwilling to annotate their images. Another issue is that words chosen to annotate an image can be highly subjective, with appropriate keywords changing between different users which in turn can render keyword-based search unintuitve and difficult to operate [39].

### 1.1 Content-Based Image Retrieval

Since textual annotations are not available for most images, searching for particular pictures becomes an inherently difficult task. Luckily a lot of research has been conducted over the last two decades leading to many interesting methods for content-based image retrieval [75, 11]. Content-based image retrieval (CBIR) does not rely on textual attributes but allows search based on features that are directly extracted from the images [75]. This however is, not surprisingly, rather challenging and often relies on the notion of 'visual similarity' between images or parts thereof. While humans are capable of effortlessly matching similar images or objects, machine vision research still has a long way to go before it will reach a similar performance for computers.

Smeulders *et al.* [75] define three primary applications of CBIR systems. A *target search* is undertaken when the user has an absolute target in mind, perhaps of an exact image or images of a specific object or scene. A *category search* is undertaken when a user requires an image that best represents some class of images. Finally, in *search by association*, users have no initial aim other than to search for images of interest. This usually leads to an iterative procedure whereby the search may be focussed on an image which the user finds interesting.

### 1.2 Query-By-Example

In the early days of CBIR, the general method used by systems such QBIC [15], Virage [20], PhotoBook [52] or NeTra [43], to query an image database was through a query-by-example (QBE) approach. QBE allows a user to specify a query image to the system in order to retrieve images from the database that are deemed similar to that query. Each image is characterised by a feature vector (e.g. the bins of a colour histogram as originally proposed in [76], or a combination of colour, texture and shape features as in [15] - see [75] for a detailed review on image features). An equivalent feature vector is extracted from the query image and compared to all database vectors to arrive at similarity or dissimilarity scores between query and database images (using metrics such as $L_1$ [76] and $L_2$ [15] norms or the earth mover's distance [64]).

Upon comparing the database images to the query, the system will present the top $N$ similar images according to their distance from the query image. The presentation of results is typically a one-dimensional linear arrangement, in order of increasing distance (i.e. decreasing similarity) starting from the top left hand corner of a grid.

There are two main drawbacks of QBE-based CBIR. The first one is that users may not deem the images presented by the system as actually being similar to the query. For example, a user may supply the system with a red flower. The system will return all images with a large red content, and the texture and shape similar to a flower. However the user may be searching either for red flowers, or a particular species of flower that happens to be red in their particular picture. This high-level interpretation of an image by the user cannot be satisfied by the low-level feature calculations performed by the computer. This problem is of course not specific to QBE-based retrieval but is common to all similarity-based CBIR approaches and is known as the 'semantic gap' [75].

The second shortcoming of QBE is that a user may not actually have an image to give to the system, thus rendering QBE effectively useless. While potential solutions such as sketch-by-example [28, 38] have been proposed in order to overcome this issue, these have limitations of their own and are hence rarely explored.

## 1.3 Relevance Feedback

A commonly explored approach to improve the retrieval performance of CBIR systems, and a partial solution to the first issue presented above, is relevance feedback (RF) [86]. This mechanism modifies the underlying parameters of the algorithms of a system in an attempt to learn what a user is searching for. Upon presentation of the initially retrieved images, the user can specify whether they deem a retrieved image useful or not. Multiple images can be selected as either positive or negative examples and these are then used in order to weight the different features according to the user's preference, and update the search results which should now contain more relevant images. This process can be repeated to further improve the retrieved results. In the aforementioned example of the red flower, if the user were to select multiple images of red flowers as positive examples the system is then likely to return more red flowers, weighing the colour feature more highly than shape or texture. On the other hand, if the user selects images of the same species of flower but with varying coloured petals, the system will emphasise shape and texture more than colour. A variety of RF mechanisms exist [86], the most common being a relevant or non-relevant selection (as e.g. used in [15]) or slide mechanisms allowing the user to specify a continuous score of relevance (as employed e.g. in [65]).

The user will generally only select a small amount of positive and negative examples. Therefore, small sample learning methods are required. The most successful of these methods include discriminant analysis and support vector machines (SVMs) [77]. In the work of Tao *et al.* [78], the authors state that SVM based RF has shown promising results in previous studies due to good generalisation abilities, but show that incorporating asymmetric bagging and a random subspace into a SVM, can lead to improved results, while reducing computational complexity. The authors of [77] experiment with variations of discriminant analysis for RF, namely LDA (Fisher linear discriminant analysis) and BDA (biased discriminant analysis) and develop an improved method named directed kernel BDA (DKBDA). The reader is directed to works such as [86, 77, 78] for further information on these and other RF algorithms.

Another variation of RF allows the user to manually drag the system results closer or further away from the query image based on preference [23]. Other browsing-based RF mechanisms are described in Section 3.5.

### 1.4 Image Browsing Systems

Image browsing systems attempt to provide the user with a more intuitive interface, displaying more images at once in order to harness the cognitive power of the human mind in order to recognise and comprehend an image in seconds. Interaction with a traditional QBE system can often lead to a confusing and frustrating user experience. Formulating queries from images can prove difficult for the user, and the 'black-box' state of such approaches means that users typically cannot derive how the system is retrieving these results, and are thus unable to modify the query in order to improve the results returned by the system.

This is confirmed in a user study presented by Rodden and Wood [62] where the authors provided users with an image retrieval system that offered a variety of querying facilities, including speech recognition and the traditional QBE approach. The authors found (by examining usage logs) that most users did not use the QBE function as the system did not meet their unrealistic expectations of the current state of CBIR. For example, a user explained how he had attempted to find all the images of a new blue car by using a query image, but the images provided were irrelevant. As he had no idea how the system was providing these results, he could not improve the query and thus abandoned the search.

Browsing systems give a useful alternative to QBE. Providing an overview of the database to the user allows for intuitive navigation throughout the system. This is particularly the case when images are arranged according to mutual similarity as has been shown in [59], where a random arrangement of images was compared with a visualisation which positioned images according to their visual similarities, i.e. where images that are visually similar to each other are located close to each other in the visualisation space. It was discovered that during a target search (i.e. looking for a particular image), similarity-based visualisation reduced image retrieval time.

QBE systems cannot be used when the user does not have a specific image in mind, as no query image can be provided. Image browsing systems overcome this problem by showing an overview of the image database. An overview of the collection will give the user a good indication whether or not an image or image class they have in mind might actually be present in the database. In some cases, the entire database will be displayed to the user on a single display. The user can then focus on regions of the visualisations that they are attracted to or believe will harbor a particular concept they have in mind. Browsing such visualisations when arranged according to image similarity, as shown in [59], can increase the rate of retrieval. These visualisations are usually achieved through dimensionality reduction, whereby the relationships between images in a high-dimensional feature space are maintained as best possible in a reduced 2D (or 3D) space which is more comprehensible to the user.

In case image collections are too large to fit to a single display, images can be grouped according to similarity through the application of a clustering procedure. The user is then able to navigate through these clustered groups of images in order to browse the collection. An overview of the database is provided by initially presenting the user with a representative image for each cluster. Clustering can also be performed in a hierarchical manner which in turn allows for visualisation of very large datasets.

Another way in which image databases can be displayed is through graph-based visualisations. In these approaches, links are formed between images that are deemed similar or that share a common concept, while the images themselves form the nodes of the graph. The whole connected graph, or part thereof, is then displayed to the user for visualisation and navigation.

Similarity-based visualisation is not the only useful form of arranging image databases. In particular for personal collections, grouping according to the time images were created has shown to be useful. This approach can be adopted to automatically cluster event images. In cases where time information is not always available or not necessarily reliable, this approach can be combined with similarity-based systems.

The fundamental issue with the development of a browsing system is how to present the user with the images in a database. With image collections ranging in the size of millions, any browsing system needs to utilise the limited screen space provided by a typical computer monitor in a manner which is intuitive and easily navigable by the common user. Immersive environments and virtual reality allow for a completely new way of visualising information with a unique user experience. It is only natural that this approach has also been adopted for visualising image databases. The user is immersed into the actual database, while the addition of a third dimension coupled with the larger visualisation space can lead to a more effective approach of navigation.

While a visualisation of an image collection is useful for providing an overview of the contained images, it provides only part of a useable image database navigation system. Once a collection is visualised, users should have the ability to interact with it in order to arrive at the image(s) they are looking for. Typical operations here include panning and zooming which allow the user to focus on images in a different part of the visualisation space, respectively on images which were previously hidden.

With regards to the three primary CBIR applications of [75], browsing interfaces clearly allow for better *search by association* (searching with no specified target) than QBE approaches. As for *target search* (looking for a particular image), QBE interfaces may provide quicker retrieval times compared to a browsing interface, but of course need a query image to start with. For *category search*, arranging images by similarity creates intuitive groupings of images relating to the same category. On the other hand, formulating a single query image of a category for QBE could prove difficult. For example, suppose a user wanted to enrich a travel article of Australia with a handful of pictures. It is not clear which images in the database would be best suited, without seeing all the Australia related pictures in the database. QBE could only work in this instance if the user knows exactly which aspect of Australia they require (e.g. an image of a kangaroo or the Sydney Opera House). In contrast, allowing the user to browse the database can help cross the 'semantic gap' by allowing the user's cognitive system to play a more active role during image selection.

Browsing systems can provide users with a much less constrained, continuous interface in order to explore an image database. In this chapter, we review a variety of methods used by different researchers in order to arrange and visualise image databases to support intuitive image database navigation. The rest of the chapter is organised as follows: Section 2 focusses on how these databases can be visualised, explaining approaches based on dimensionality reduction, clustering, and graph-based visualisations. Section 3 describes different tools implemented by researchers in order to enable users to browse these visualisations. Section 4 highlights user studies undertaken in the field, how they are performed and what discoveries such studies have found. In each section we provide a critical discussion of the various approaches proposed in the literature. Our observations are summarised and future directions identified in Section 5.

## 2 Visualisation of Image Databases

In order to browse an image database, the users need to be presented with thumbnails of the images so that they may intuitively navigate the database. The primary issue associated with visualisation is how to best display the images within the limited space of (typically) a 2D screen. A variety of methods have been devised in order to visualise images, whether it be the entire database or a subset of images. In this section we look at the different techniques used in order to visualise image databases for browsing.

### 2.1 Mapping-Based Visualisation

CBIR systems typically employ high-dimensional features to represent images. Clearly, it is impossible for the human mind to perceive a feature space of this magnitude, and based on the raw data, we are therefore unable to recognise potential relationships within the dataset. In order to visualise this high-dimensional data, various techniques exist which describe the feature space layout within a low-dimensional model which the human mind can more readily understand. For image database browsing, this mapping is typically down to just two dimensions, namely the $x$ and $y$ co-ordinates of a 2D computer display. The main problem is obviously how to perform this mapping so that the relationships of the original data are maintained. In the following, we discuss various approaches that have been employed to this effect.

### Principal Component Analysis (PCA)

Principal component analysis (PCA) is the simplest dimensionality reduction approach, working in a linear manner. The starting point for PCA is the symmetric covariance matrix of the feature data from which the eigenvectors and their respective eigenvalues are calculated and ranked in descending order of eigenvalues. The principal components are selected from the top eigenvectors according to the number of dimensions required (i.e. for 2D the top two eigenvectors are selected). These eigenvectors are then used to plot the original data where image thumbnails are plotted at the co-ordinates derived through projection of the orginal feature data into the low-dimensional space. PCA has the advantage that it is relatively simple. However, since it maximises the variance of the captured data it does not necessarily best preserve the mutual relations between the individual data items (this is only the case if the underlying metric in the original feature space is the $L_2$ norm).

The Personal Digital Historian (PDH) project developed by Mitsubishi Electronics Research Lab (MERL) [45] uses PCA splats in order to visualise images. PDH attempts to bring photo sharing to a round table top, with the system being projected down from above. The authors use colour, texture, and shape features which are then projected, using PCA, to a 2D format whereby similar images appear close together. Keller *et al.* also use a PCA visualisation to present images in a virtual 3D interface based on texture features [31].

### Multi-Dimensional Scaling (MDS)

In contrast to PCA, multi-dimensional scaling (MDS) [36] attempts to preserve the original relationships (i.e. distances) of the high dimensional space, as best possible in the low-dimensional projection. MDS starts with a similarity matrix which describes all pair-wise distances between objects in the original, high-dimensional space. The goal is then to best maintain these distances which in turn can be formulated as minimizing a ‘stress’ measure, often defined as [36]

$$\text{STRESS} = \frac{\sum_{i,j}(\hat{\delta}_{ij} - \delta_{ij})^2}{\sum_{i,j}\delta_{ij}^2} \tag{1}$$

where $\delta_{ij}$ is the original distance between objects $i$ and $j$, and $\hat{\delta}_{ij}$ is the distance in the low-dimensional space. Starting from either a random initial configuration, or from the co-ordinates after applying PCA, the algorithm continues to reposition the images in order to reduce the overall stress, until a termination condition has been reached (for example a maximum number of iterations or threshold stress value).

MDS was employed by Rubner *et al.* [64] who suggested using it for browsing image collections. Based on colour signatures of images and the earth mover's distance (EMD) [64], the authors were able to create a representation of the high-dimensional feature space using MDS, placing image thumbnails at the co-ordinates derived by the algorithm. Figure 1 shows an example of a MDS visualisation of an image database.

**Fig. 1.** An MDS visualisation of the UCID image database [74]

MDS provides a more accurate representation of the relationships between images in the feature space compared to PCA. The work of [64] also suggested that MDS can be used for both image query results (local MDS) and to give an overview of a collection of images, providing the user with a general scope of images contained within the database (global MDS). However, MDS comes at the cost of more expensive computation compared with PCA, working in quadratic time. This suggests that image co-ordinates cannot be calculated interactively, and thus that MDS is not well suited to present query results. For global MDS, though image co-ordinates may be calculated off-line in order to browse the data set interactively. Additional difficulties arise when adding images to a collection visualised through MDS, as this typically requires recalculation of the whole dataset and the relocation of image thumbnails in the visualisation.

Rodden *et al.* have investigated the use of MDS for image database visualisation based on the evalution of several user studies [59, 60, 61, 62].

In [59], they compare two approaches, one based on random assortment of images, and one using a similarity-based MDS interface, and conclude that the MDS-based system is faster for locating specific images.

MDS has also been used to measure the effectiveness of particular feature vectors for conveying similarity within a CBIR system. In [40], MDS is employed to manually inspect the similarity derived by using the MPEG-7 texture retrieval descriptor (TRD) and texture browsing descriptor (TBD). They conclude that using the TRD with either the $L_1$ norm or EMD distances provides more suitable MDS layouts. Besides visual inspection, they also used spatial precision and recall in order to arrive at quantitative conclusions.

These accuracy measures, which are adaptations of the classical precision and recall measures used in information retrieval, were first proposed in [58], where a quantitative comparison between different distance measures is undertaken to examine which provides the best MDS visualisation according to similarity perceived by humans. In order to calculate the average spatial precision and recall, each image in the database is treated as a query image. In Figure 2 the dashed circles represent the increasing levels of recall from the query image (coloured dark gray). The levels of recall are set based on the next closest relevant image (coloured light gray) to the query. The number of relevant images within a circle is divided by the total number of images in that recall circle to calculate spatial precision. This is then averaged for all the recall circles, giving an average level of precision.

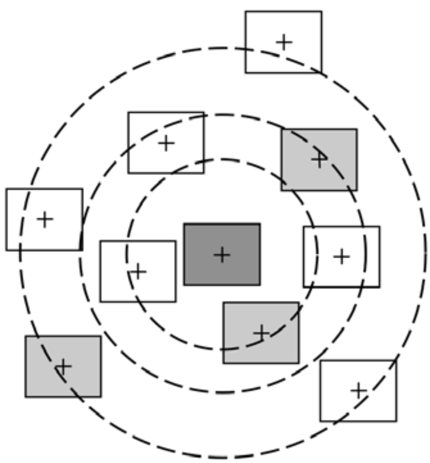

**Fig. 2.** Illustration of the spatial precision and recall measures used in [58]

Using these measures, [58] examines the quality of visualisations when using different indexing methods and distance measures. They evaluated feature vectors consisting of averages of hue, saturation and value, localised average hue, saturation and value features (where the image is partitioned into 9 regular grid cells), and colour signatures as used in [64], HSV histograms using the $\chi^2$ (chi-squared) and Jeffery Divergence measures, and finally a scheme named IRIS, which is a fairly complex index introduced in the paper. The authors first compared the indexing techniques using a standard QBE system, where results show that the more complex IRIS indexing method achieves the best precision and recall. They then explored how these indexing techniques compare in terms of average spatial recall and precision for an MDS visualisation. Interestingly, they found that here the simplest measure, namely average HSV values, is able to retain roughly 85% of the accuracy, whilst IRIS achieves only around 52%. The authors conclude that, in a reduced dimensionality space, an average HSV MDS visualisation is comparable with a more complex indexing technique, such as IRIS, yet much more simple to compute. They furthermore investigate the computational complexity in more detail and report that the most time consuming indexing technique is the colour signature/EMD method of [64] which takes about 230 times longer to compute a full similarity matrix compared to the average HSV computation.

### FastMap

FastMap is an alternative dimensionality reduction technique devised by Faloutsos and Lin [16]. FastMap is able to reduce high-dimensional spaces down to a linear 2D or 3D space. The algorithm selects two pivot objects, an arbitrary image and its furthest possible neighbour. All points are mapped to the line that connects the two pivot points using a hyper-plane located perpendicular to the line that connects the two pivots. The co-ordinates where images appear on the hyper-plane can be used to display the images on the screen, maintaining the relationships which occur in the high-dimensional space. As with MDS, a distance matrix is required as input for the algorithm.

The advantage of FastMap is that it requires less computation compared to MDS, having a linear $O(kn)$ complexity, where $n$ is the number of images and $k$ is the number of dimensions to reduce the data to. In their experiments, the authors tested FastMap against MDS, showing more than comparable results in much shorter times. This suggests that FastMap could potentially be used for computing visualisations 'on-the-fly', for example to visualise results of QBE searches. The resultant visualisation however, is not always as accurate as those created by MDS.

FastMap is also employed in the virtual reality system 3D MARS [47] to map images to a 3-dimensional space in which users can virtually navigate themselves around the image database through query selection (see Section 2.4 for more details on virtual reality visualisation systems).

### Self-Organising Maps

A self-organising map (SOM) [33] is a neural network which is trained to perform feature extraction and visualisation from the input of raw data.

Using an input layer of neurons, the feature vector of a sample is computed and assigned to a best matching unit (BMU) on a 2D map. Each unit has an associated weight vector, with the same dimensionality of the feature vectors computed from each of the samples in the dataset. A learning rule, typically defined as

$$w_i(t+1) = w_i(t) + \gamma(t)h_{b,i}(t)[x(t) - w_i(t)] \tag{2}$$

where $w_i(t)$ is the weight vector of node $i$, $\gamma(t)$ is the learning rate and $h_{b,i}$ is a function modifying the weights around the BMU, is then applied to update the weight vectors.

Applied to image databases, employing SOMs leads to similar images being located closer together on the resulting 2D map than less similar images [12]. To avoid a time consuming linear search of what could be an extremely large map (according to the size of the database), hierarchical self-organising maps (HSOMs) can be constructed where only root BMUs need to be compared to the input vector during mapping [12].

An earlier use of SOMs for image database visualisation is the PicSOM system [37]. PicSOM uses layers of parallel SOMs to form a hierarchy, in particular a tree-structured self-organising map (TS-SOM) [34]. Here also, a linear search of all units in the map for the BMU of a given feature vector (constructed in PicSOM using MPEG-7 descriptors for colour, texure and shape) is avoided, by restricting the search for a BMU to a $10 \times 10$ unit search below the BMU of the previous level. This reduces the overall BMU search complexity from $O(n)$ to $O(\log(n))$. After training has been implemented on each of the TS-SOM levels (using each image in the test set 100 times), each node is assigned the image most similar from the database. This results in similar images being mapped closer together than dissimilar images on the 2D map. These representative images may then be browsed by the user in a hierarchical manner (see Section 3.2).

While the work of Zhang and Zhong [85] focusses on the development of a content-based HSOM as an indexing structure, Deng *et al.* [12] and Eidenberger [14] implement visualisations that facilitate the browsing of the images in the database. Deng *et al.* [12] train a HSOM using Sammon mapping [67], an MDS variant. The low-level features extracted from the images were regional CIEL*u*v* colour averages, an edge density histogram and texture features extracted through Gabor filters. In their experiments, the system was used to visualise a collection of 3,000 images.

Eidenberger [14] describes a system where HSOMs of video stills are created based on a variety of MPEG-7 descriptors. Each input vector is compared with a BMU representing a cluster of images. When these clusters of images are visualised by the HSOM, a representative image (closest to the BMU weight vector) is displayed. Furthermore, an HSOM is employed for time-based visualisation. Here, each node is required to be visualised by exactly one image, rather than a cluster as with similarity-based visualisation. This is achieved by using the weight vector of each node in the map and assigning it with the closest image feature vector available in the database. The output maps were computed on a hexagonal layout, and images cropped to hexagons.

### Other Mapping-based Techniques

A range of more recent techniques for visualising high-dimensional data are investigated by Nguyen and Worring [49]. The three non-linear embedding algorithms employed are ISOMAP (isometric mapping), SNE (stochastic neighbour embedding) and LLE (local linear embedding). In ISOMAP [79], nearest neighbour graphs are formed within the data, and the shortest path between every pair of points is calculated, with the length of the path being used in a distance matrix for MDS. SNE [26] calculates the probability that any two points take each other as nearest neighbours in both the high- and reduced-dimensional space, and attempts to match the two probability distributions. LLE [63] can be seen as an approximation of SNE. The authors further propose to merge ISOMAP with SNE and LLE to form two new techniques, ISOSNE and ISOLLE. In ISOSNE, the distances found through ISOMAP are used to form the probabilities used by SNE, rather than using MDS, and ISOLLE is derived in an analogous way.

In their evaluation they found that both ISOSNE and ISOLLE perform better than MDS. Although ISOSNE performed best, the computation time was reported at being around 10 times that of ISOLLE. The authors therefore concluded that if off-line calculations can be performed, ISOSNE can be used, while for faster visualisations, ISOLLE should be the method of choice.

Milanese *et al.* [44] describe the use of correspondence analysis [30] as a dimensionality reducing mapping technique. Using a data table, a mathematical function is applied in order to create an observation matrix, which can be be used with the eigenvectors of a covariance matrix in order to project the data table into the 2D space. This formulation allows both images and features to be projected onto a common space, and to distinguish which features are closer to a particular cluster of images.

### Handling Overlap in Visualisations

Rodden *et al.* [59] observed that the vast majority of users do not like the overlapping and occlusion effects occurring in MDS displays due to images being located too close to each other (see also Figure 1). This issue with partial or even total occlusion is of course not exclusive to MDS, but also occurs in other visualisations such as PCA splats. Co-ordinates that are close together in the feature space will inevitably become even closer in a 2D representation generated through mapping. When image thumbnails are overlaid at these co-ordinates, parts of or indeed entire images are hidden from the user.

In order to combat this, various systems invoke some mechanism which adapts the layout in order to reduce the amount of overlap occurring between images. Much work here has focussed on mapping the visualisation to a regular grid structure. Gomi *et al.* [18] used MDS "as a template" in order to locate images within rectangular regions representing a cluster. Rodden *et al.* [59] developed a method for spreading the images around a grid. First the co-ordinates are used to locate the ideal grid cell for an image. Should this cell be already occupied, a spiral search emanating from the selected cell is performed in order to locate the closest free cell (see Figure 3 on the left). In addition to this basic strategy, where the image is simply mapped to the next closest free cell, a further *swap* strategy was also proposed. Here, an image is moved to the next closest cell, and the new image is placed in the optimal cell. Finally, in a *bump* strategy, the images in the line of cells between the optimum cell and the next closest cell are all moved outwards (from the optimum centre cell) by one cell, with the new image being placed at the centre optimum cell. From experiments it was found that the *bump* strategy produces the lowest average error (i.e. lowest average distance an image is from its optimal cell). The complexity of the algorithm is $O(m^2) + O(n^2)$ where $m$ is the size of the grid and $n$ is the number of images to be located. The three strategies are presented visually in Figure 3 on the right. This technique was also adopted by Schaefer and Ruszala in [73] and [71] to spread out images on an MDS plot and a spherical visualisation space respectively.

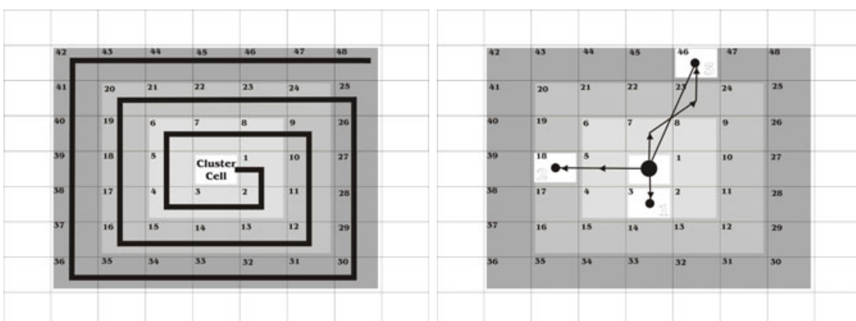


**Fig. 3.** The spreading strategies proposed in [59]

Liu *et al.* [42] developed two different approaches for overlap reduction in order to present web search engine results. Their first technique also fitted the visualisation to a grid structure, but they comment that the *bump* strategy of [59] works in quadratic time, and is thus not suitable for real-time use. Their method creates an ordered data set, optimised in one dimension while sub-optimising the other and has a complexity of $O(2nlog(n) - nlog(m))$ where $m$ is the number of columns or rows and $n$ is the number of images. Their second technique allows the user to dynamically set the amount of overlap through use of a slider bar. Image co-ordinates are established by

$$P^i_{new} = \gamma P^i_{Sim} + (1-\gamma) P^i_{Grid} \tag{3}$$

where $P^i_{Sim}$ and $P^i_{Grid}$ represent the locations of the image in the similarity-based and grid-based visualisations respectively, and $\gamma$ is the overlap ratio controlled through the slider bar.

Nguyen and Worring [49] specify two requirements with regards to dimensionality reduced visualisations, a *structure preservation requirement* and an *image visibility requirement*. The first requirement states that the structure of the relationships between images in the feature space should be retained, while the second demands that images should be visible enough so that the content of the image is distinguishable. It is clear that these two are intrinsically linked. Moving an image in order to make it more visible will detract from the original structure, while maintaining the structure could cause a loss of visibility in certain images.

As a solution to this, Nguyen and Worrring define a cost function which considers both image overlap and structure preservation. In order to detect overlap, a circle is placed about the centre of the image, as it is assumed that an object of focus will be about the centre of an image. If the circles of two images overlap, the position of the images will be modified according to values derived from the cost function. A similar cost function is also used to modify the PCA visualisations in the PDH system of Moghaddam *et al.* [45].

### Discussion

From the various works that have employed mapping-based techniques, it is clearly difficult to formulate a direct comparison of which is best. Each individual approach uses a different image database and different underlying features and distance measures to quantify the similarity between images. Ruszala and Schaefer [66] attempt to compare PCA, MDS and FastMap by considering the complexity of the algorithms required. They conclude that if accuracy is of importance then MDS should be used, otherwise FastMap should be implemented when faster visualisation generations are required. However this study does not include the more recent use of local linear embedding algorithms detailed in [49], shown to be faster and as accurate as MDS. Future work could aim at comparing a variety of dimensionality reduction visualisations using the spatial precision and recall measure defined in [60, 58]. The use of approximation algorithms such as FastMap and LLE, operating at lower complexity than more accurate algorithms such as MDS, offers the possibility of visualising dynamically produced data sets such as query results.

From works such as [59, 42, 49] it is clear, that image overlap is an undeniable problem for users who prefer to see images in their entirety. Much research has been undertaken into how is best to resolve this problem. Moving images too far from their mapped location can cause the relationships in the full-dimensional feature space to be distorted, and hence there is a trade-off between image clarity and maintaining the overall structure of the relationships [49]. The placement of images within a grid structure is a visualisation which the general user is familiar with. Hence, arranging images within a grid according to their mutual similarities will typically enhance the general user's browsing experience.

## 2.2 Clustering-Based Visualisation

Dimensionality reduction techniques applied to image database visualisation are fundamentally limited by the number of pixels displayed on a computer monitor, as this will directly determine the number of images that can be displayed on the screen. Much work has been undertaken in order to reduce the number of images to be displayed to the user at any one time. This is usually achieved by clustering groups of similar images together, so that only a single image for each group is displayed to the user, hence freeing up visualisation space. In this section we describe the principle methods in which images can be grouped automatically for the purpose of image database visualisation, and how each group can be portrayed by representative images.

### Content-based Clustering

Content-based clustering uses extracted feature vectors in order to group perceptually similar images together. The advantage of this approach is that no metadata or prior annotation is required in order to arrange images in this manner, although image features or similarity measures which do not model human perception well, can create groupings that may potentially make it difficult for a user to intuitively browse an image database.

Krischnamachari and Abdel-Mottaleb [35] were among the first to propose clustering images by image content. Local colour histograms (extracted from image sub-regions) were used to cluster similar images and each cluster was visualised using a representative image. Schaefer and Ruszala [72] also cluster images based on colour descriptors (the average hue and value in HSV colour space).

Hilliges *et al.* [25] use a combination of colour, texture and roughness features. These are extracted based on a YUV colour histogram, some Haralick texture features and the first four roughness moments of the image. The resulting clustering is utilised in conjunction with an image quality classification technique. The work by Borth *et al.* [5] represents another example of content-based clustering using colour and texture features.

K-means clustering is one of the most commonly used clustering techniques which iteratively approximates cluster centres. Image database navigation approaches that employ k-means include the works by Abdel-Mottaleb *et al.* [1] and Pecenovic *et al.* [51]. Hilliges *et al.* [25] use a variant of k-means named X-means. In their approach, images are first clustered using colour histograms comprised of the u* and v* values of the images in CIEL*u*v* colour space. This way, the system is able to detect series of multiple similar images, which are then classified based on image quality in order for users to only keep their best photographs.

### Metadata-based Clustering

Despite the difficulties of manually annotating images, work has been undertaken to visualise images according to this associated metadata. The introduction of systems such as ALIPR (Automatic Linguistic Indexing of Pictures - Real Time) [39] demonstrates that images can be automatically annotated. However, this assignment of high-level semantic meaning by machines is still in its infancy and often not very reliable. Systems such as ImageGrouper [48] and EGO (Effective Group Organisation) [81] allow the user to manually arrange images into clusters and perform the bulk annotation of the contained images.

CAT (Clustered Album Thumbnail) by Gomi *et al.* [18] uses a combination of keyword and content-based clustering. At the top level of the clustered hierarchy, images are clustered by keywords. The user is presented with a list of keywords, of which they can select one or more. Upon keyword selection, all images in the database associated with the chosen keyword(s) are clustered by localised average colour content (average CIEL*u*v* values from grid cells placed over the image) and image texture (calculated through a Daubechies 4 wavelet transform). Each cluster takes a representative image, which in higher levels is sized dependent on the proportion of images from the database that are located in that particular cluster. At lower levels of the structure, images are arranged more uniformly in a grid-like structure using MDS and PCA templates.

The rectangular boxing of clusters employed is similar to that used in PhotoMesa. PhotoMesa [4] has the ability to arrange images in quantum treemaps or bubblemaps. Quantum treemaps are designed to display images of indivisible size regularly, whereas bubblemaps fill the space with indivisable items but generate irregular shaped groups. Images with a shared metadata attribute (e.g. directory, time taken, or keyword) are grouped together. When an image is first loaded into the database, multiple sized thumbnails of the same image are stored in a filesystem and dynamically loaded based on the size of the rectangular sections.

For the quantum treemap algorithm, the input is a list of numbers specifying the size of the rectangles, and the display space. The output is the layout of the rectangles. The algorithm generates rectangles with integer multiples of a given element size, where all the grids of elements align perfectly. When images are assigned to their groups, an evening algorithm is run to re-arrange the images in the boxes. The authors note that a relatively large amount of wasted space may occur on the screen, particulary when the number of images in a group is small. PhotoMesa has three different grid arranging mechanisms in order to irradicate irregular layouts. The size of the rectangle is dependent on the proportion of images from the database that cluster contains. Figure 4 shows an example of a regular layout of images in PhotoMesa.

In an attempt to remove unused space, [4] also introduces the idea of using bubblemaps in order to visualise the database. In this approach, images are

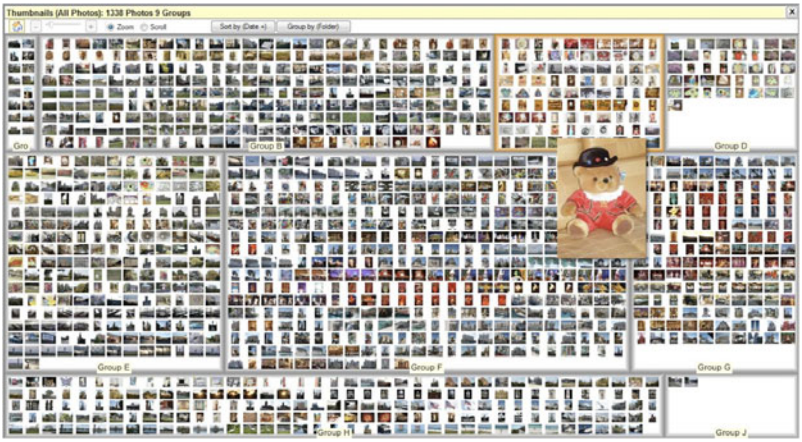


**Fig. 4.** Clustered images, taken from the UCID dataset [74], visualised as a quantum treemap in PhotoMesa [4]

still displayed on a regular grid, but the surrounding area can be arbitrary in shape.

### Time-based and Combined Time/Content-based Clustering

Time-based clustering uses time stamp information associated with an image in order to group images within a collection. This time data may have been provided either by the digital camera when the photograph was taken, or by an operating system when the image was euploaded from the camera or downloaded from the internet, or set manually by the user. The possible ambiguity of when a time stamp may have been attached to an image can indeed be the downfall of this particular method of grouping. Furthermore, some images may contain no time stamp information at all [55].

It has been demonstrated by Rodden and Wood [62] that users find browsing through time-ordered images more intuitive than content-based browsing (see also Section 3.4 for more discussion on this). Graham *et al.* [19] justify their approach of grouping and visualising images according to time with the observation that *"people tend to take personal photographs in bursts"*. Based on this premise, images are clustered according to the time difference between time stamps with images first being clustered by year, then month, day then hour. The authors give an analogy of a birthday party in order to explain sub-clusters in their approach. The event itself will take up an entire day, but different parts of the day may contain different bursts of images, for example blowing out the candles on the cake may have several images attributed to it.

The time-based clustering algorithm employed in Calendar Browser [19] is based on Platt *et al.*'s PhotoTOC system. PhotoTOC (Photo Table of Contents) [55] visualises images in two panes: overview and detail. In the overview pane, a grid of representative images is presented to the user, arranged by month and year, where each image corresponds to a particular time cluster. Images are arranged in a sequential list and new events can be detected by

$$log(g_N) \geq K + \frac{1}{2d+1} \sum_{i=-d}^{d} log(g_{N+1}) \quad (4)$$

Assuming $g_i$ is the time gap between image $i$ and image $i+1$, $g_N$ is considered a gap if it is much longer than the average gap. $K$ is an empirically selected threshold value and $d$ is the window size.

The main difference between the approach by Graham *et al.* and PhotoTOC is how identified events are sub-clustered. In [19], medium sized clusters are first created by a pre-defined time gap. These new clusters are then sub-clustered by the rate at which images are taken for that cluster. This rate can then be matched with the other intra-cluster rates to split and merge clusters. Parent clusters are developed through fixed measurements of time, i.e. events that occurred in the same day, week, month and year. In contrast, PhotoTOC sub-cluster events based on colour content, therefore not completely relying on time stamp information. This approach is hence also applicable to image sets which are only partially time-stamped.

Another approach using a combination of time- and content-based clustering is the PhotoSim system [8]. PhotoSim uses k-means to cluster images already clustered via time, enabling the system to derive clusters that model human perception. For this, they utilise colour histograms based on the U and V components of the YUV colour space. In the example shown in Figure 5, the images in the cluster have been separated into portraits and pictures of buildings taken at either night or day.

### Hierarchical Clustering

Hierarchical clustering can be seen as analogous to file structures found in common operating systems with clusters of images corresponding to folders and individual images being mapped to files. Indeed, this is often how users organise their personal collections. The majority of systems that cluster images, arrange clusters in a hierarchical manner. Examples of this can be found in [55, 8, 18, 5].

Hierarchical clustering algorithms are typically divided into agglomerative and divisive methods [29]. Agglomerative, or bottom-up clustering, begins with treating each individual sample as an individual cluster. Using a form of similarity, clusters are merged with their most similar neighbours and this process is repeated until a pre-defined number of clusters remain. These clusters then form the top layer of the generated tree. In contrast, divisive, or top-down, clustering begins with all samples starting as a single large cluster which is then iteratively split into smaller clusters until a termination criterion is met (such as all clusters corresponding to individual samples). In

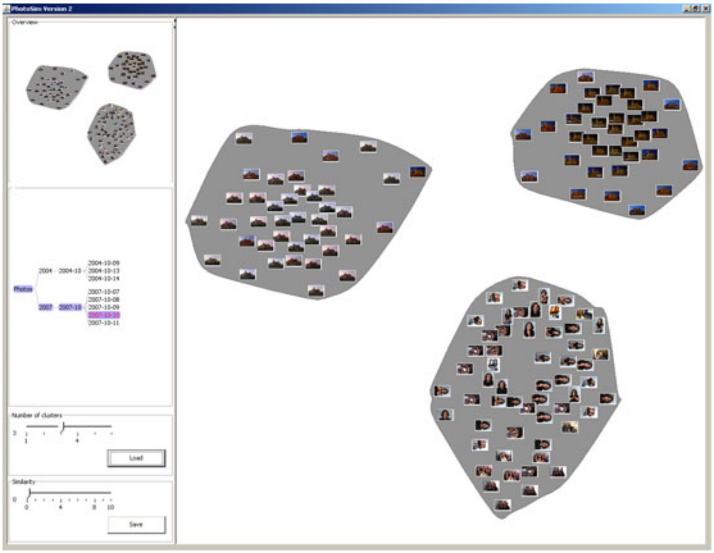

**Fig. 5.** Images from a cluster in PhotoSim [8] further clustered into portraits, buildings at night and buildings in the day

terms of image database visualisation, the leaf nodes of the tree correspond to the individual images, while the nodes at different levels of the tree form the various image clusters.

Despite being computationally more expensive than partional methods such as k-means clustering, it has been shown that approaches based on agglomerative clustering afford better retrieval accuracy [1]. Krischnamachari and Abdel-Mottaleb [35] use local colour histograms to form a hierarchical structure of images. First, each image is treated as its own cluster, and represents a leaf node of the tree. From all the clusters, the two with the most similar average colour histograms are merged together to form a parent cluster. Consequently each parent node has exactly two child nodes, forming a binary tree.

The CAT system in [18] first uses agglomerative clustering to group images initially by the keywords associated with them, and then creates internal clusters based on colour and texture features, again through the application of agglomerative clustering. Borth *et al.* [5] also use agglomerative clustering for their Navidgator system, which allows browsing through a dataset of video stills.

Pecenovic *et al.* [51] employ a hierarchical form of k-means clustering, where nodes are successively split as proposed in the LBG algorithm [41] to form a tree structure that can be visualised and browsed.

A hierarchical structure can also be derived without the application of an actual clustering algorithm. This is demonstrated by Schaefer and Ruszala in [73] and [72] who perform a uniform quantisation type clustering based on the definition of a grid structure for visualisation. Once the grid is defined, each image in the dataset will fall into one of the grid cells. Each grid cell hence corresponds to an image cluster. A spreading algorithm as in [59] is applied to reduce the number of unused cells and the number of images assigned to one partiular cell. When multiple images are mapped to a particular cell, a tree structure is formed by subdividing each cell into further uniform partitions with the spreading algorithm being applied to the root grid and all child grids in order to prevent the addition of unnecessary levels in the hierarchy. Based on this structure, using a grid of $24 \times 30$ cells and an assumption that 40% of the cells are assigned images, the system could visualise $((24 \times 30) \times 0.4)^3$, i.e. over 23 million images.

**Selection of Representative Images**

For visualisation purposes, each clustered group of images needs to be represented either by a single image or perhaps a small group of images. The manner in which these representative images are selected can vary between systems. In many approaches (such as the one in [72]), the centroid image of the cluster is selected. Formally, this is the image with the minimal cumulative distance from all other images in the database. Alternatively, other systems such as CAT [18] select the image closest the the centroid of the cluster in the feature space. A similar approach is adopted in PhotoTOC [55]. Here, to derive the most representative image of a cluster, the Kullback-Leibler divergence between every image histogram in the cluster and the average histogram for all images in the cluster is measured. The image with the colour histogram closest to the average histogram of the cluster is selected to be the representative image.

A cluster may also be visualised using more than one representative image. For example, the clustered visualisation of web search engine results generated by Liu *et al.* [42] displays a cluster preview of 4 images. Another content-based representative image selection scheme with the ability to display several representative images is that of Krischnamachari and Abdel-Mottaleb [35]. Based on a user-defined number of representative images $R$, a set of representative images $R_n$ is formed. If $R = 1$, then the representative image is the leaf node of the sub-tree with the feature vector closest to the average feature of all images in a conjoined set $R_0$. When $R > 1$, image selection is taken from several subsets of images. Referring to Figure 6, if the user requires $R = 2$ representative images for cluster 14, the subsets will be $R_0 = 1, 2, 3, 4$ and $R_1 = 5$. The image most similar to the average of $R_0$ will be selected, together with the sole image from $R_1$.

While the works of [18, 55, 72] use content-based analysis in order to select a representative image for the cluster, the Calendar Browser in [19] chooses representative image(s) based on time. The system displays a summary of 25 images at any granularity (i.e. year, month or day). This 25 image summary

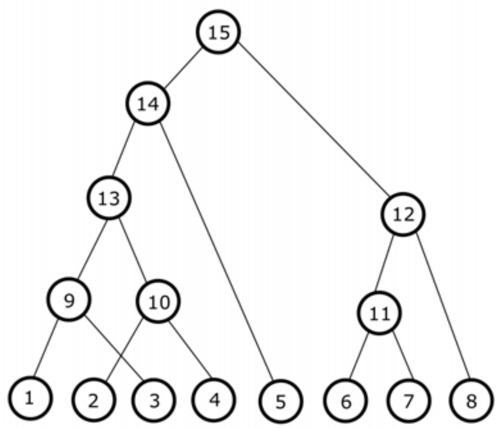


**Fig. 6.** Example of the hierarchically clustered image database arranged as a binary tree in [35] (© 2009 IEEE)

is created using a two step process. The first step is screen space assignment, where one space is assigned to each cluster. If there are too many clusters, priority is given to large clusters. Any remaining spaces after the allocation of a single space to each cluster are divided amongst the clusters according to their size. This creates a target number of photos for the second step, which performs the actual selection of representative images. The first criteria for selection is based on consecutive images with the smallest time difference, since it is likely that images taken close together describe the same event. One of these two consecutive images is then selected as the representative. If from the first step more than one representative images are required, the largest time difference between images is used, which will typically signify a new event and the second image in this pair will be selected.

### Discussion

Clustering-based visualisations have the advantage that a user is given an overview of all images contained within the database at the top level of the hierarchy without displaying each individual image. This gives a good summary of the database. In addition, clustering can be performed in a hierarchical way leading typically to a tree structure representation of the database. As the user traverses this tree, the images become more similar to each other, and hopefully also more suited to the type of image the user is browsing for [5]. One downside of this approach is that if an image is erroneously clustered by the system (i.e. is assigned to a cluster of images that are not very similar to it), it will make that particular image very difficult for the user to locate, effectively making it lost. An example of this would be searching for an image of children playing football in a park, in a database that clusters based on colour similarity. In such a system the image in question might be clustered together with images of plants due to the green colour of the grass the children are playing on. If the chosen representative image is then also one of a plant, intuitively the user may not think to navigate into that cluster.

## 2.3 Graph-Based Visualisation

Graph-based visualisations utilise links between images to construct a graph where the nodes of the graph are the images, and the edges form the links between similar images. Links can be established through a variety of means including visual similarity between images, or shared keyword annotations. Once a graph has been constructed, it needs to be presented to the user in a visualisation that allows for intuitive browsing of the database.

### Mass Spring Visualisation

Dontcheva *et al.* [13] use a mass spring model to generate the visualisation. A spring is formed between two images if they share an associated keyword. The length of the spring is assigned based on the number of images sharing the same keyword and a constant used to control the density of the layout. To generate the layout, the visualisation is evolved using the Runge-Kutta algorithm [3]. The authors conclude that this technique is only suitable for relatively small databases of a few hundred images due to the time required to stablise the arrangement. Worring *et al.* [83] also created a mass spring visualisation [6] based on keyword similarity (the number of keywords a given pair of images have in common). A $k$-nearest neighbour network is then formed based on this similarity measure. In order to visualise this high-dimensional structure in 2D, connected images are placed closer together while unconnected images are moved further apart. This is achieved by applying attractive or repulsive forces respectively between the images. The authors claim that this visualisation technique aids particularly when implementing a category search (i.e. searching for an image of a particular class), due to the fact that an image selected by the user will have nearest neighbours most relevant based on keyword. For example, selecting a picture of a cat with an associated keyword "pet" could present the user with images of dogs, cats or any other domesticated animal. A set of user interactions are available, designed to reduce the amount of effort required to form a subset of purely relevant images (see Section 3.1). User simulated tests were performed using only relatively small visualisations of up to 300 images. The main difference between the visualisations of [13] and [83] is that in the system of Worring *et al.*, the links between images are explicitly displayed whilst Dontcheva *et al.* do not display such links.

**Pathfinder Networks**

The use of Pathfinder networks [6] for image browsing was introduced in [7]. The interface, fronting an image database named InfoViz, was used in conjunction with QBIC [15], allowing the user to query and browse the database. Pathfinder networks were originally used to analyse proximity data in psychology, although many other types of high-dimensional data can also be represented using this technique [6]. The underlying theory behind Pathfinder networks is that a link between two items exists if it is the shortest possible link. The Pathfinder algorithm removes all but the shortest links by testing for triangle inequality. In the case that this does not hold, the path is considered redundant and is removed from the network.

For the layout of the network, images with many links between them are considered similar and therefore placed closer together, while images with fewer links are generally located further away. Chen *et al.* inspect the visualisations produced using colour, texture and layout features from the images and state that colour (through use of a colour histogram) provides the best visualisation, achieved using a spring-embedder node placement model. Figure 7 shows an image database visualised using colour histograms in such a Pathfinder network, where images with similar colour histograms form clusters. The experiments in [7] were implemented on a database containing 279 images. With such a small image collection, it is difficult to predict how well Pathfinder network visualisations may scale to larger image database sizes.

**Fig. 7.** An image database visualised using a Pathfinder network based on colour histograms [7]

### $NN^k$ Networks

$NN^k$ networks, where NN stands for nearest neighbour and $k$ describes a set of different features, were proposed by Heesch and Rüeger [22] to browse through an image database. The basic principle is that a directed graph is formed between every image and its nearest neighbours if there exists at least one possible combination of features for which the image is the top ranked of the other. Seven different features were extracted, including an HSV colour histogram, a colour structure descriptor (for detailing spatial formation of colour), a thumbnail feature (where the image is scaled down and gray values calculated), three texture features and a 'bag of words' (stemmed words taken from text attributted to images) feature.

A weight space is used which is a pre-defined set of weights for each of the features. The number of weight sets for which an image is top ranked, forms the similarity measure between images. For example, assuming that three weight sets are defined together with a query image $Q$, then if image $A$ is ranked top in the first image set, but image $B$ top in the second and third weight sets, image $B$ will take a higher proportion of the weight space and therefore is deemed more similar to $Q$ than image $A$.

Each image in the network stores its nearest neighbours, along with the proportion of the weight space for which the image is ranked top. Given a query image, a user-defined number of nearest neighbours will be displayed to the user, as well as links between the neighbours. Images with a higher similarity (i.e. a higher proportion of weight space) are displayed closer to the query which is centralised on the display. The initial display to the user is an overview of the database, generated by clustering the images and displaying the most representative thumbnail from that cluster (as described in Section 2.2). Figure 8 shows an example of how the network is visualised after an image has been selected as a query. In their experiments, the authors used a database containing 34,000 video stills.

Heesch and Rüeger [23] also describe their system's ability to query a database through keywords. In the example of Figure 9, searching the database with the query "airplane taking off" returns a variety of results. The top matching image is placed at the centre of the interface, with the nearest neighbors placed along an Archimedean spiral according to the proportion of the weight space they possess in terms of the query image. Images closer to the centre of the image are larger in size than those on the periphery of the spiral. The user can drag these smaller images closer to the centre where they are dynamically resized and can be inspected more closely by the user. They may then select multiple images to further the query.

### Discussion

The use of graph-based visualisations appears to be less common than either mapping-based or clustering-based visualisations. Graph-based visualisations

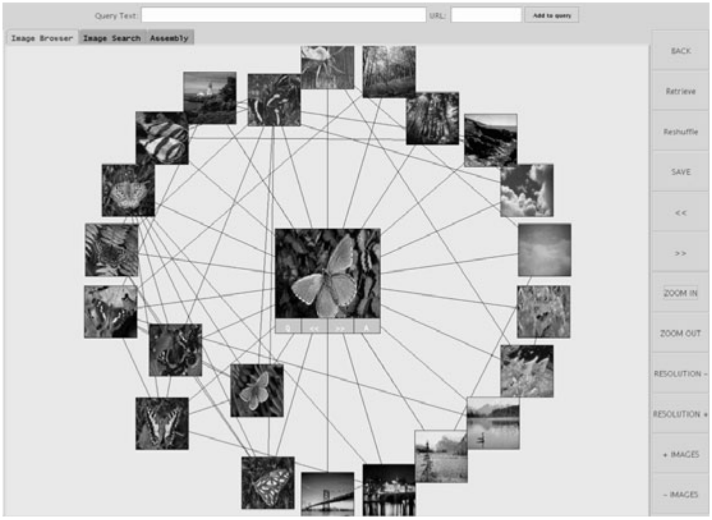


**Fig. 8.** An example of a an NN$^k$ query selection taken from [22]

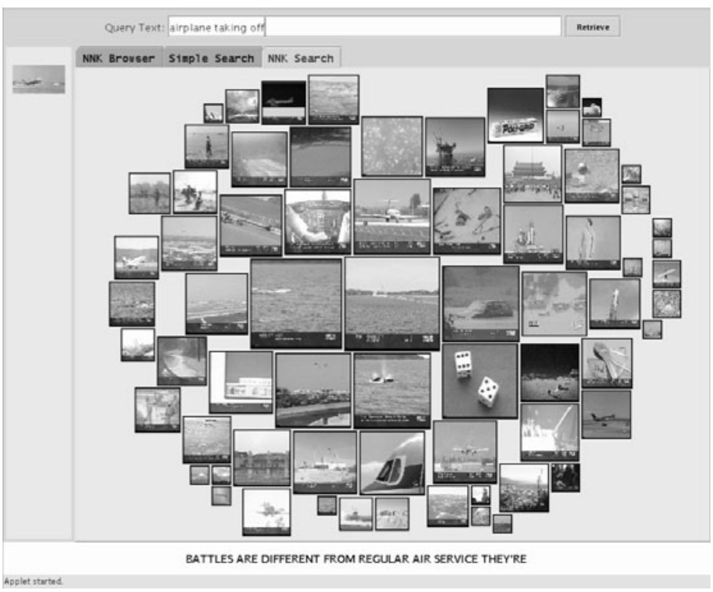


**Fig. 9.** An example of a query for "airplane taking off" in the interface devised in [23]

are typically quadratic in complexity, and therefore can only be computed off-line in order to allow for real-time browsing. Generating query results 'on the fly' is not particularly suited to this style of visualisation. As with dimensionality reduced or clustered visualisations, the introduction of additional images in the database often requires re-calculation of the entire structure.

The major approaches in graph-based visualisations use contrasting visualisation methods. While mass-spring models and the Pathfinder network present a global visualisation similar in form to that of mapping-based techniques, $\mathrm{NN}^k$ visualisations present images one by one, allowing users to make an interactive choice on the next image to pursue. This is closer to traditional QBE methods, although the implementation of similarity by proximity should aid the user more than a linear format. Whilst $\mathrm{NN}^k$ networks can have a vast number of links (dependent on the size of $k$), Pathfinder networks attempt to minimise the number of links between images. So far, no study has been undertaken to explore which graph would allow for faster retrieval through browsing. It would also be of interest to see how well Pathfinder and Mass Spring networks are able to visualise larger databases, such as that used for testing the $\mathrm{NN}^k$ network.

## 2.4 Virtual Reality-Based Visualisation

The development of image browsing interfaces has also produced some interesting approaches based on the use of virtual reality (VR) equipment and software. Rather than limiting the user to traditional input hardware such as mouse and keyboard, work has been conducted using more interactive devices such as head tracking equipment [82] and the use of input wands [82, 47]. In general we can divide VR-based image visualisation techniques into two classes: immersive and non-immersive visualisations.

The 3D Mars system [47] visualises an image database in 3 dimensions. Images are projected onto four walls (left, right, front and floor) of a CAVE environment [10] around a user wearing shutter glasses. The interaction with the system begins with a random assortment of images from the database. As the user moves between the walls, the images rotate to face the user in order to prevent images being hidden. A virtual compass is provided on the floor allowing the user to 'fly' through the 3D space. The user can select a starting query image from the random assortment via use of an interactive wand. Each of the dimensions in the display represents either colour, texture or structure features. The query image selected by the user is placed at the origin of the axis, and all similar images are visualised in the space dependent on their distance from the query image. This visualisation is generated through the application of FastMap, as described in Section 2.1. The novelty of the system is that it makes interaction much more interesting for the user. However, unfortunately the system does not have the functionality to visualise an overview of the entire database.

In [82] the authors present their system StripBrowser. Images are arranged upon filmstrips and can be ordered using colour content along a rainbow scale or from light to dark. A user can navigate along each filmstrip using a head-tracking device (see also Section 3.1). An issue with projecting all images along one dimension is that the user must browse each image sequentially. This will require more user interactions compared with various approaches that use a 2D or 3D visualisation space.

Non-immersive VR image browsing systems create a virtual environment for users to navigate around in order to view the images in the database. Tian and Taylor [80] use MDS to plot 80 coloured texture images in a 3D space. The images are wrapped to spheres and plotted at the locations derived by MDS based on features vectors comprising a PCA projection of a colour and texture histogram. The user can navigate through the 3D space using a control panel located at the bottom of the screen. An issue not tackled by this system though is the potential overlap of spheres, presumably not occurring within the small database used for testing.

A different non-immersive VR image browsing system is presented by Assfalg *et al.* in [2]. Here, a graphical environment allows the user to move around a virtual world taking photographs of scenes in order to query the database. Upon loading the environment, pre-defined shapes are randomly placed within the scene. The user can 'walk' through the environment using navigation icons located on a panel on screen. The user may then edit the objects in the environment, having the ability to add new pre-defined shapes to the current scenery, and to texture and colour the shapes as desired. Shapes in the prototype system presented include a variety of tree like structures, statues and buildings. The user can select a rectangle over the current view in order to take a photograph, something the authors state as an intuitive metaphor for the user. The selected portion of the scene within the rectangle is used as a starting point for an adjoined QBE system, which retrieves all similar images from the database. Textures can be taken from results retrieved through QBE and applied to the environment in order to achieve modified results. Figure 10 shows a set of possible interactions a user may have with the browser.

## 3 Browsing Image Databases

In the previous section of this chapter we looked at how the often large image databases can be visualised and presented to the user. Although usually closely related, browsing the database is not the same as visualising it; Webster's dictionary defines browsing as *"to look over casually"* and visualisation as *"putting into visible form"*. A variety of tools have been developed which aid the user in order to interactively browse the images in a database. In this section we review common tools included within image database navigation systems to aid in the task of ultimately arriving at images of interest in an effective and efficient manner. We divide browsing methods into horizontal

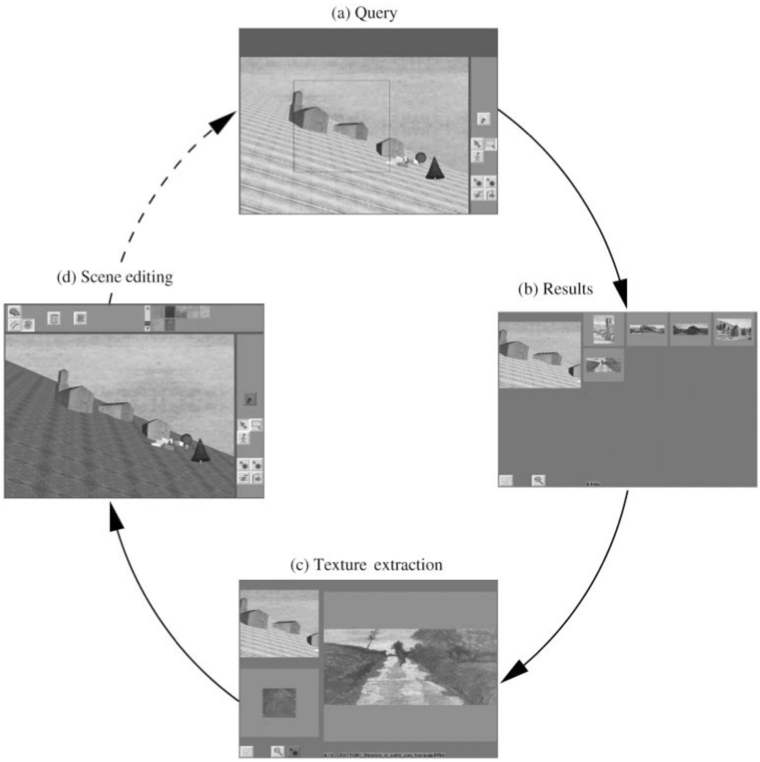


**Fig. 10.** Interactions available in the VR browsing environment presented of [2]

browsing, which presents images on the same level of a visualisation to the user, and vertical browsing, which can be used to navigate to a different level of the collection. Graph-based visualisations are typically browsed by following the links between images. For systems that organise images based on time stamps, browsing methods should also take this information into account. Finally, browsing can also be usefully employed in relevance feedback mechanisms.

### 3.1 Horizontal Browsing

We can define horizontal browsing as the navigation within a single plane of visualised images. This type of browsing is often useful when an image database has been visualised either through a mapping scheme (as described in Section 2.1), a single cluster of images (Section 2.2) or through a graph-based visualisation (Section 2.3). Several tools have been developed in order to support this browsing experience.

### Panning

If the entire visualised image collection cannot be displayed simultaneously on screen, a panning function is required in order to move around the visualisation. There are a variety of different ways in which panning can be implemented within browsing interfaces. The simplest manner in which a user can pan through an image collection is through the use of traditional scroll bars. This is particularly the case when images are arranged in a regular grid format, such as in the QBIC interface [15]. If possible, scrolling should be limited to one direction only in order to reduce the number of actions required by the user to browse the entire collection [54]. An alternative to scroll bars is the use of a control panel for panning (and zooming) as implemented in various approaches. The systems described in [7, 72, 80] all provide such a navigational toolbar enabling the user to browse through the visualisation space.

The hue sphere system by Schaefer and Ruszala [72] allows for intuitive panning by the user, as it uses the metaphor of a globe in order for users to browse the images in the collection. Images are plotted along the latitude of the globe according to the average hue of the image, whilst the average value is used to plot the image upon the longitude of the globe. The user is able to spin the globe about either horizontal or vertical axes, in order to bring images into view. This is illustrated in Figure 11, showing an image collection after various rotation/panning operations by the user.

StripBrowser [82] also allows for intuitive panning using a head tracking device. As the user looks to the right side, images are being scrolled to the left, while looking to the left causes the scrolling of images to the right. The greater the angle at which a user moves, the faster the scroll motion will be. Scrolling only occurs when the angle reaches a threshold value, otherwise the strip remains stationary. The 3D Mars system [47] allows users to pan the generated visualisation by walking around the 3D space projected on the four CAVE walls.

### Zooming

When presenting many images on a single 2D plane, the thumbnail representations of images often have to be reduced to small rectangles which are difficult to distinguish on the screen. This can be seen in the MDS plot in Figure 1. There, although it is possible to see that the images vary in colour, it is not possible to depict the content of each individual image. It would therefore be useful to have a facility to zoom into an area of interest.

For dimensionality reduced visualisations, Rubner *et al.* proposed zoom operations on a global MDS visualisation, using of a joystick in order to *"get closer to the area of interest"* [64]. Another example of a dimensionality reduced visualisation with a browsing interface facilitating zooming is the CIRCUS system presented in [51]. CIRCUS uses multi-dimensional scaling, in particular Sammon mapping [67], in order to present representative images

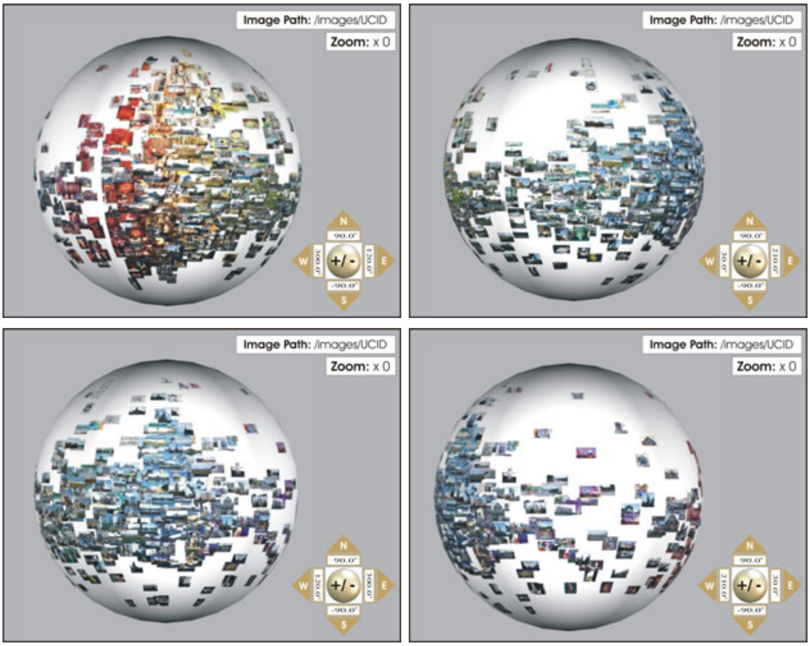


**Fig. 11.** Browsing through UCID images [74] of different hues in the system proposed in [72]

at each level of the hierarchically clustered database (where clustering is based on content). By selecting a 'Browse Collection' tab, users are presented with a browsing window which, at the minimum zoom factor (i.e. zoomed out as far as possible), shows the Sammon mapping layout of all images at that level of the database.

To reduce the amount of computation required, and to allow the user to browse the database interactively, CIRCUS displays images at the minimum zoom factor simply as dots. This maintains the user's understanding of the relationships between the representative images at this level of the database, while reducing the amount of processing time required by the system. As the user zooms into an area of interest, thumbnails are rendered. When the user focusses on a single image, metadata associated with that image is also presented. Further zooming on a particular image causes CIRCUS to present the next level of images in the hierarchy and hence implements vertical browsing (see Section 3.2).

CIRCUS also implements what is described as a *"fixed small overview"*, preventing the user from becoming lost in a 2D space larger than that of the display area. A detail view is provided, where the images are shown together with an overview displaying a map of the overall visualisation (with

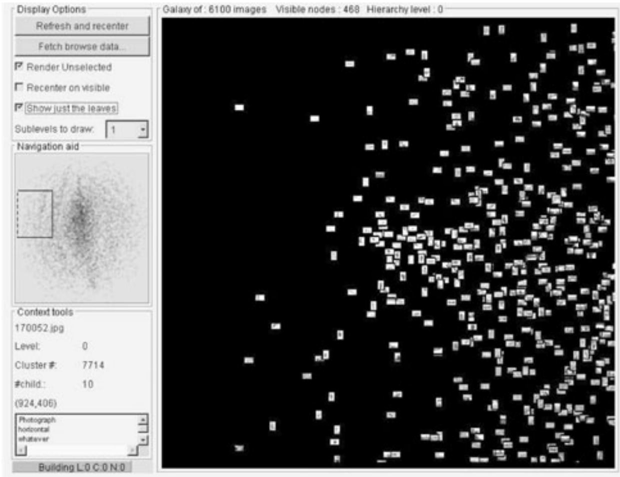


**Fig. 12.** A screenshot of the CIRCUS browsing interface presented in [51]

the current area of focus highlighted). This is presented to the user in the left hand pane of the CIRCUS browser, shown in Figure 12.

Another system using a similar overview approach is the Photosim system presented in [8], allowing users to view and modify multiple clusters (described in more detail in Section 3.5). A zooming tool is also implemented in the hue sphere system devised by Schaefer and Ruszala [72].

While most zooming interfaces require the use of a computer mouse, a more novel approach to zooming is adopted in the StripBrowser system [82]. Zooming in and out is achieved by moving closer and further away respectively from the screen. The authors note that this is an ideal metaphor for users, as generally to inspect an item in the real world a person will move closer to it. Another implementation of this metaphor is provided in the fully immersive 3D Mars system [47], where the user can zoom in on an image by physically moving closer to the wall on which it is projected.

Hilliges *et al.* [25] provide a clustered visualisation with a zoomable interface. The user may zoom into particular clusters to examine the images within them more closely. In the graph-based system by Chen *et al.* [7] a control panel located at the bottom of browser window has two control buttons allowing the user to zoom the current view in the detail pane in or out. A very similar interface is also implemented by Tian and Taylor [80], allowing the user to zoom in and out of a 3D MDS visualisation of textured images.

## Magnification

Although similar to zooming, magnification usually occurs when a cursor is placed over an image. This maintains the overall structure of the visualisation by rendering only small thumbnails for each image in the database at first, while higher resolution images are loaded only when required. An example of a system using mouse over magnification in order to dynamically display a higher resolution image is PhotoMesa [4], illustrated in Figure 4.

The hexagonal browsing system by Eidenberger [14] also provides a high resolution preview image when the mouse cursor is moved over any of the images in the visualisation. For the system created for the user studies conducted by Rodden *et al.* [61], a 3x magnification of an image occurs when the cursor is placed over an area of the MDS visualisation.

Another form of image magnification that can be used for examining image database visualisations is the application of a fisheye lens [69]. Using this magnification mechanism, images located at the centre of the lens are magnified whilst those immediately around the focused image(s) are distorted [54]. Figure 13 shows an example of how a fisheye lens could effect a collection of images fitted to a regular grid.

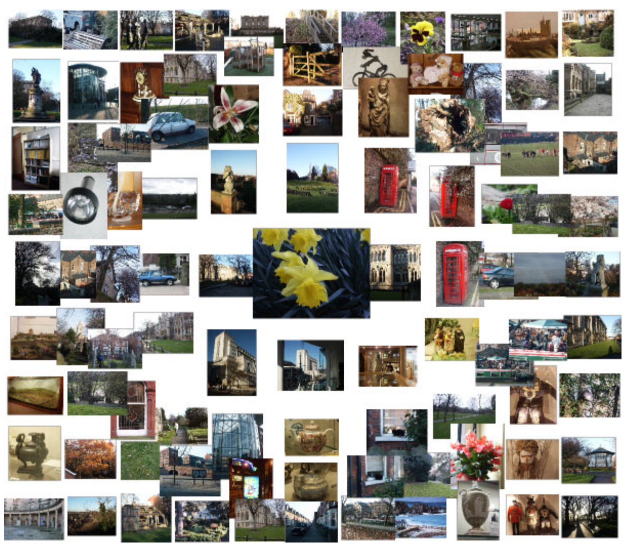

**Fig. 13.** Example of a fisheye lens browsing over images from the UCID dataset [74]

## Scaling

Some browsing systems use scaling, rather than zooming, allowing users to view a particular image in more detail. In the EIB (Elastic Image Browser) system [57], the user may use two slider tools in order to dynamically resize images both horizontally or vertically. This enables the user to display more images in the browser, at the expense of image clarity. Images can also be portrayed as lines, with the colours in the lines becoming the only distinguishing feature between images. The author claims that this could potentially speed up browsing. However, in the EIB visualisation, images are not arranged by mutual similarities; rather they are placed randomly within the grid visualisation leading to a negative effect in terms of the user's browsing experience.

The PDH system [45] also includes a slider tool so that the user may dynamically resize images, whilst maintaining the 2D spatial relationships between images achieved through PCA.

### 3.2 Vertical Browsing

In visualisation approaches that are based on a hierarchical structure, the contained images can also be navigated using vertical browsing methods. As discussed in Section 2.2, clusters of images are typically visualised through the use of representative images. These images are crucial for vertical browsing as they are typically the reason for which a vertical browsing step into the next level of the hierarchy is initiated by the user.

In the quantum treemap visualisation of image clusters provided in PhotoMesa [4] (shown in Figure 4), the user may click on a highlighted image in order to invoke a smooth zoom into that group of images. The box around the selected cluster remains highlighted to prevent user confusion. Zooming may continue until a single image is displayed at full resolution. The CAT system [18] provides a functionality similar to this.

In systems using a regular grid structure at different levels, such as the hierarchical hue sphere by Schaefer and Ruszala [72], or the hexagonal browsing system by Eidenberger [14], selecting a representative image at a given layer of the hierarchy will present the user with the subsequent layer of the subtree, for which the selected image acts as the root. The user may traverse all layers of the tree. The hue sphere system also displays a visual history of the followed browsing path, while in the hexagonal browsing system, the user is presented with a view of both the previous layer and a preview of the layer described by the currently selected cell. A similar combination of history and preview is included in the PhotoMesa system [4].

A navigational history is also provided in the Navidgator video browsing system [5] shown in Figure 14. Here, the user's most recent image selections are displayed in the top right hand corner of the interface, and may be revisited by selecting one of the thumbnails. The representative images at each level are displayed in the lower portion of the screen. Selecting an image creates a larger preview just above the layer viewing portion of the interface, and also adds a thumbnail of the image to the history. The user may then *zoom* in or out of the levels in the database using arrow buttons. Single arrows move the user up or down a single layer of the database (up to the previous layer, or down to the first layer of the sub-tree for which the currently selected representative image acts as the root) while double arrows enable the user to perform a *multi-level zoom*, whereby every third layer of the tree is displayed. A *max zoom* function is also included which allows the user to navigate directly to the bottom or top layers of the tree.

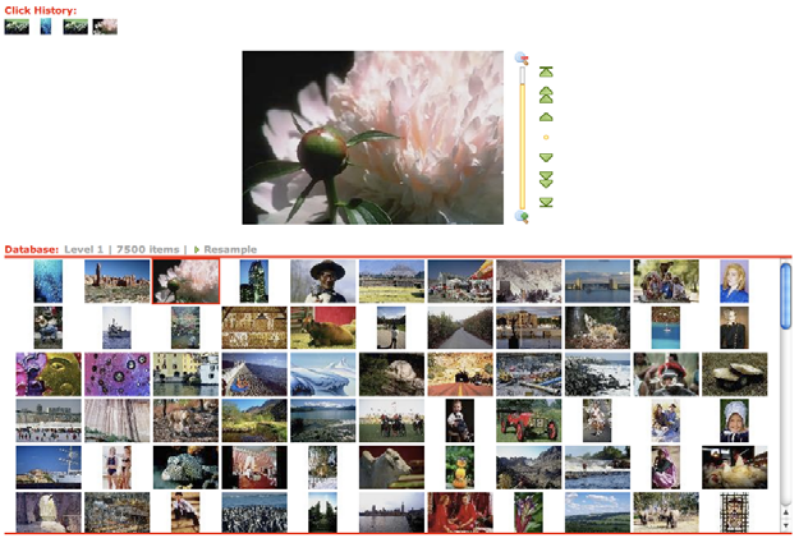


**Fig. 14.** A screen shot of the Navidgator system detailed in [5]

A different vertical browsing function is implemented in the CIRCUS system [51]. The authors introduce a semantic zoom facility which allows users to zoom into areas of interest. As the user zooms into a representative image beyond a certain zoom factor, the sub-clusters associated with the cluster of interest are automatically displayed.

### 3.3 Graph-Based Browsing

Operations such as panning and zooming can also be applied to graph-based visualisations. For example, in the Pathfinder network approach by Chen *et al.* [7], a global view of the network is presented (as shown in Figure 7). Displaying this global representation of the structure bears some similarity with some of the mapping-based visualisations of image databases from Section 2.1. In the Pathfinder system, a toolbar is displayed at the bottom of the browsing window, which the user can use to zoom into areas of interest or to pan around the collection. Images found through browsing may then be selected in the interface to be used as a query for the QBIC system [15].

The structure of the graph itself however also allows for different methods to browse from image to image. This is realised in the NN$^k$ network approach by Heesch and Rüeger [22] which exploits the links between images in the graph. First, the user is presented with an overview of the database through representative images of clusters formed through a Markov chain clustering. The user can then select one of these images as a query image. As shown in Figure 8, the selected image is placed at the centre of the screen and a user defined number of nearest neighbours are placed around it based on their similarity to the query image. Furthermore, links between these neighbours are also displayed. Selecting a neighbour will put it as the query image in the centre, with its nearest neighbours then presented in a similar fashion. This process can then be repeated until a required image has been found. The user also has the ability to zoom in or out of the visualisation. As with a typical web browsing interface, a 'Back' button is provided so that the user may return to the previous query, should the newly selected image provide no images of interest.

Browsing the graph-based visualisation developed by Worring *et al.* [83] lies somewhere between the two browsing methods of the $NN^k$ and Pathfinder visualisation styles. While the overview of the database is presented, the user may invoke one of five different actions in order to select a subset of images that are deemed relevant. A user may select a single image, as well as selecting the single image and all of the linked neighbours in the network in a single action. The opposite two actions are also available, whereby the user can deselect a single image (and disable it from future automatic selections) or an image and the associated neighbours. Another action available to the user is the ability to expand the current selection of images by automatically selecting all connected neighbours. A simulated user test showed that the provided interactions can reduce the amount of effort required to select all possible relevant images (compared with selecting images one-by-one). Worring *et al.* conclude that using the functionality of selecting an image and all the nearest neighbours, followed by deselecting all images deemed irrelevant, a higher recall and precision measure can be achieved whilst maintaining the same interaction effort required for a one-by-one selection technique.

### 3.4 Time-Based Browsing

As described in Section 2.2, time stamp information attached to images can be used to cluster and visualise image collections. Clearly, if a collection is visualised based on temporal concepts, browsing should also be possible in a time-based manner. One of the earliest time-based image browsing systems is the AutoAlbum system introduced in [56] further developed into the PhotoTOC system [55]. Here, a two-level hierarchy based on time is utilised. As can be seen in Figure 15, dates, in monthly intervals, are shown in the overview pane on the left hand side of the interface, with the representative images of the clusters falling into that date also being displayed. Selecting a representative image displays the contents of that cluster in the detail pane, located to the right of the interface.

Whilst AutoAlbum and PhotoTOC restrict the user to monthly intervals, the Calendar Browser in [19] allows the user to 'zoom in' to other time intervals by selecting one of the representative images. At the year level, two controls located at the top of the interface provide a summary of the previous year and next year respectively. This approach is also adapted for the case

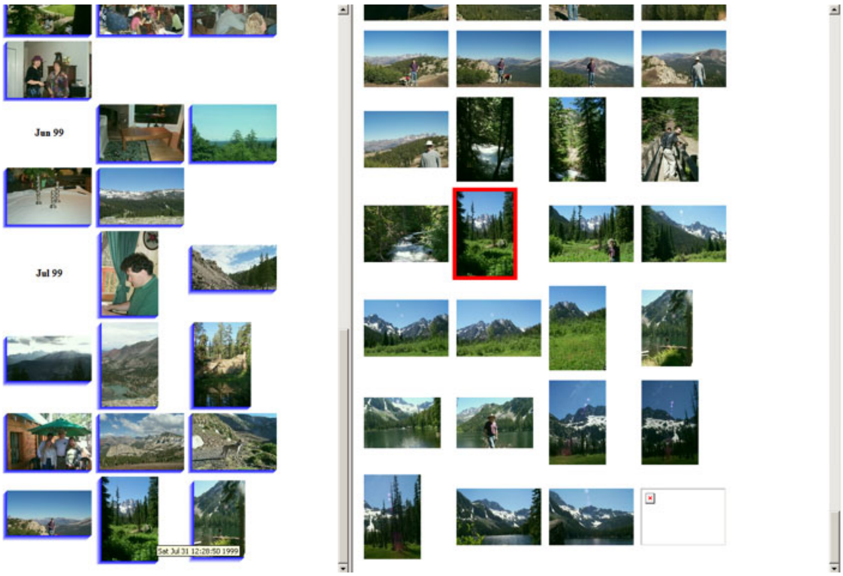


**Fig. 15.** The PhotoTOC [55] interface. The user has selected the image bottom and centre of the overview pane (left). This image has been highlighted in the detail pane(right), and is amongst visually similar images.

when viewing images at a monthly granularity (i.e. summaries of the previous and next month are displayed). When viewing images with a time stamp attributed to a particular day, images maybe browsed 25 at a time. Selecting an image at this level places it in the centre of the interface, with the images taken immediately before or after displayed around the selected image.

In [19], a modification of the Calendar Browser is also implemented and tested. In the modified interface Hierarchical Browser, a pane located on the left hand side displays a hierarchy of dates. Starting at root nodes representing years, these can be expanded to display monthly nodes, followed by dates and time intervals. Selecting a node from this pane displays the representative images in the detail pane located on the right side of the interface. This is similar to the approach of PhotoTOC [55]. User testing suggested that users could use the Calendar Browser more quickly, but the number of task failures occurring was lower in the Hierarchical Browser.

A different approach to time browsing is presented in the PhotoHelix system [24]. An interactive touch screen table top is used with an interactive pen and a specially developed piece of hardware created using the workings of an optical mouse and an egg timer. By placing the hardware on the interactive screen, a virtual helix is created at the location of the hardware. Images are arranged on the helix according to time, with newer images being located closer to the outside of the spiral. Grouped images, known as piles, are magnified when the spiral is rotated under a fixed lens. The magnified group can then be manipulated by the user through use of an interactive pen. New groups can be formed or individual images scaled, rotated and moved freely around the interactive screen.

The table top PDH system [45] also allows users to browse images according to time. By selecting a 'Calendar' button, images are sorted along a linear timeline.

The hexagonal browsing system of Eidenberger [14] allows users to swap between a content-based and a time-based tree structure. As the time-indexed tree has all the key frames of the collection visualised (as described in Section 2.1), any cell selected in the content-based tree will have a correseponding cell in the time-based tree. However in the content-based tree, images may only occur as leaf nodes if they have not been selected as representative images for clusters. Therefore, when switching between an image in the time-based tree to the content-based tree, the leaf node of the corresponding cell is selected and a message is displayed to the user in order to minimise confusion.

### 3.5 Browsing-Based Relevance Feedback

As described in Section 1.3, many CBIR systems use some form of relevance feedback (RF) in order to tailor the search towards the current user's needs. The most common mechanisms are the standard relevant/non-relevant classifier, used in QBIC [15], and a slider tool whereby images can be given a continuous score of relevance by the user, as demonstrated by the MARS system [65]. However, the introduction of novel image database visualisations have also led to the development of new RF mechanisms.

In the PDH [45] and the El Niño [68] systems, the intrinsic weightings of feature vectors are modified by allowing the user to manually specify where images should reside in the visualisation. PDH provides the user with a small subset of images to be placed as they wish on a *"user guided display"*. Based on the user layout, PDH uses the location of images in order to estimate feature weights for colour, texture and structure. Using these weightings, a larger image collection is then presented based upon their provided layout. Figure 16 shows a user guided layout on the left, and an automatic layout of a larger set on the right.

The El Niño system [68] allows users to manipulate the entire visualisation, rather than just a subset as in PDH. Images presented to the user may be moved to modify the internal weightings of the system. Each image manually relocated by the user is considered an anchor. The distances between anchors are then used to modify the colour, texture and shape feature weights. The visualisation is then updated based on the new similarity measure. As only a subset of images is shown to the user (typically 100-300), the updated visualisation may lose images which were not selected as anchors by the user. A possible issue with these systems is it may not be clear to the user

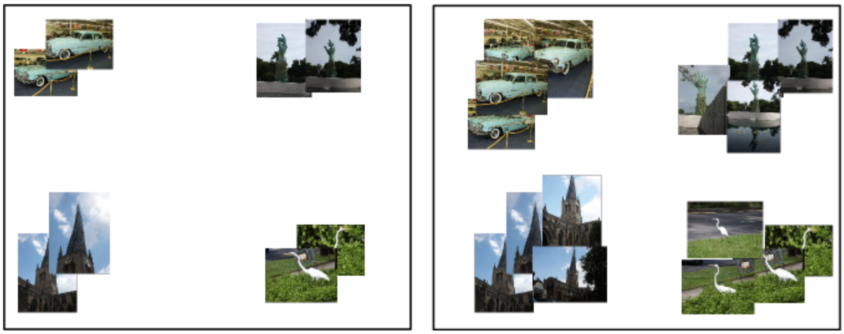

**Fig. 16.** A user guided layout of UCID images [74] in PDH (Personal Digital Historian) [45]

how far relocate images in order to modify the system to meet their search requirements [48].

Another category of mapping-based visualisations with an example of an RF implementation is the self-organising map based system PicSOM [37] (described in Section 2.1). Here, the user selects images as either relevant or irrelevant. The images, and their user determined relevance, are projected to SOM surfaces in order to find regions of relevant or irrelevant images. A low-pass filtering system is used to expand the regions of relevance on the map. A qualification value is assigned to each image based upon the relevance of the image and surrounding images. Each SOM is searched for the top 100 images with the highest qualification value. The top 20 images from the combined set are returned to the user, from which the process can be repeated if necessary.

While the above systems use RF within a dimensionality reduced visualisation, there have also been clustered visualisations with integrated RF mechanisms. An example of this is Photosim [8], shown in Figure 5. Whilst higher level clusters are created based on time, images within the same time period are clustered on content. Photosim allows users to transfer images between clusters manually, if they are not satisfied with the automatically formed groupings. Furthermore, the user also has the ability to create entirely new clusters. Using a slider tool, the user can alter the degree of similarity in which images are automatically added to the new cluster. Setting the slider to zero creates a cluster with just a single image, dragged from an existing cluster. The higher the threshold value, the degree of similarity required in order to add new images to the cluster is lowered.

Similar approaches with solely manual clustering occurs in the EGO [81] and ImageGrouper [48] systems. In EGO (Effective Group Organisation), a manually created grouping of images retrieved through some search (such as QBE, or keyword-based search) can be defined. EGO then recommends other images in the system by treating each image in the group as a positive training example, and modifying the feature weights to find similar images in the database. ImageGrouper adopts a different approach in that manually created groups can be selected as positive, negative, or neutral examples. The system will then return images based on the positive examples given by the user. Sub-groups can be made within groups in order to narrow the search. For example within a group of car images, the user may narrow the search by selecting only the red cars in the group as positive. The manual groupings in these two systems allow for *bulk annotation*. Instead of labelling each image individually, the user may simply annotate the entire group with keywords, in order to facilitate future keyword searches.

The fully immersive 3D Mars system [47] also has a RF mechanism incorporated, allowing the user to choose positive or negative examples using an interactive wand. The system then modifies the weightings of the features used to query the remainder of images in the database.

### 3.6 Discussion

The browsing tools described in this section aim to aid the user during the navigation of an image database. While horizontal browsing can be applied to all visualisations where either a selection or all images in the database are displayed to the user on a single plane, vertical browsing is limited to hierarchically organised visualisations. The user is able to select a representative image to view a collection of images similar to their selected image. In this way, the user is presented with a subset of more similar images relating to their intended target. However, unlike horizontal browsing, once the user traverses down a particular path of representative images they can lose the overview of the database. Therefore, to reduce the user's cognitive load and to minimise confusion, systems will often give the user some indication of their current position within the database. An example of this is implemented in the Navidgator system [5], whereby a textual description includes the current level of the database being displayed and the total number of images in the current layer. A potential improvement to this would be a visual map, displaying the user's current location in the database.

The two contrasting styles of graph-based visualisations are providing an overview of the collection (as in the Pathfinder network of Chen *et al.* [7] or the approach by Worring *et al.* [83]) or presenting singular images in the database and their linked neighbours (as implemented in the $NN^k$ network of Heesch and Rüeger [22]). Whilst such an overview may be explored in a similar manner to mapping-based visualisations, the $NN^k$ implementation presents a selected image as a query centralised on the display, with its nearest neighbours displayed around it at distances based on similarity. Unfortuntately, this technique suffers from a drawback similar to that of vertical browsing. Once users enter the database (from selection of an initial query image from an overview formed of representative images of clusters from the database), they may become lost in the network. The only option available to the user is to use a 'Back' button to return to previous query selections or to the initial overview. Presenting the user with the entire network visualisation prevents this problem. The user interactions presented by Worring *et al.* [83] show that such browsing techniques can reduce the amount of user effort required to create a subset of solely relevant images. However, such visualisations have only been tested with up to 300 images and it is not clear how well they would scale for larger datasets.

Time-based browsing as implemented in a variety of systems, such as Calendar Browser [19] or PhotoTOC [55], are typically aimed at personal users, as they can recall the event at which an image was taken in relation to other events in the collection [62]. Browsing such systems assumes that the images in the database are correctly time stamped, which may not be the case for all image collections.

The development of browsing systems has also resulted in some interesting relevance feedback mechanisms in which the user can dynamically update the intrinsic similarity measure by moving the position of the image directly in the visualisation space. However, it is not clear to the user how much effect a particular movement may have upon the system [48]. The EGO [81] and ImageGrouper [48] systems require the user to manually form groups before suggesting to the user possible matches. The approach undertaken in Photosim [8] automatically creates initial clusters before allowing the user to modify them, an approach that typically requires less user effort.

## 4 User Evaluation of Image Database Navigation Approaches

As is apparent from the previous sections, a lot of research has been conducted aimed at providing intuitive navigation interfaces for users of image collections. Unfortunately, the systems most widely used in practise do not offer any of these approaches. Most users rely solely on a graphical interface of file structure browsers included in common operating systems, whilst others use commercial software such as Apple's iPhoto [27] or Google's Picasa [53] in order to display their personal photos. Professional photography agencies employ staff whose sole responsibility is to manually annotate images with keywords or free text, yet they also do not employ any of the techniques reviewed in this chapter. The low uptake of browsing systems is further hindered by the fact that traditionally only few examples of image management software invoke some use of CBIR techniques, and that CBIR itself still has major challenges to overcome. In this section we explore in more detail the various tasks for which image database navigation systems are particularly useful, reducing the time required to perform them, and review various user evaluation studies that support this argumentation.

### 4.1 User Tasks

Image browsing systems can be used for a variety of tasks and tests which we will highlight in the following. Typically, different approaches are more suited to particular tasks. For each task, specific data can be extracted in order to measure the performance of a developed system. In addition, the subjective opinion of users can also be measured.

#### Target Search

Target search [75] is the most commonly employed and tested task in browsing systems, and used in works such as [9, 59, 55, 19, 18]. It is also often used as a method of testing traditional CBIR systems [46]. In target search, the user is shown an image and asked to browse the system in order to locate this target image. When the user has found the image, they perform some test termination action (e.g. click on the target image in the system). The time taken for the user to locate the image can then be recorded for further analysis. A timeout is also often implemented (i.e. when a user is unable to find the image within a specified time limit). Clearly, the gathered timing information can be used to a compare different systems, or to compare some system against a traditional search through a linear list of images.

A variation of this task was used in [48] where users were shown a target image as before, but rather than locating that particular image, were asked to select 10 semantically relevant images from the collection. Apart from the timing information, another measure that can be derived in this test is the error rate, which counts the number of images incorrectly selected by users.

The advantage of a target search task is that it is relatively easy to conduct, and enables a quantitative comparative analysis of two or more systems. It is also more likely to model the more general use of a system, e.g. browsing personal photographs.

#### Journalistic Task

As outlined in [75], a common use of image retrieval systems for journalistic purposes are *"searches to illustrate a document"*. To replicate this within a user study, participants can be given a short piece of text in which they are instructed to find a set of images from the database which best represent the topic of the text. In the experiments conducted by Rodden *et al.* [61], ten graphic design students were asked to compare an interface with images arranged according to their visual similarity through MDS assigned to a grid structure, and an interface which grouped images according to keywords, namely the geographical location of where the image was taken. Users were issued a travel article based upon some tourist destination (such as New York) and were instructed to browse 100 images from that location and select three that they deemed the most appropriate to accompany the article.

Another study by Rodden *et al.* [61] compared a similarity-based approach with a randomly arranged image set, employing a test population of average computer users. The creators of EGO [81] conducted a similar study with general users.

Graham *et al.* [19] employed a modified version of the journalistic task, providing users with a textual description and 3 minutes to locate as many images in the database relevant to the description. While comparing presentation of image from web search engines in [42], the authors asked the users in the study to find those images out of the top 200 image results that best represent the query terms.

The journalistic task, models a true requirement of a retrieval system. An issue with this test however is, that is difficult to recruit users that would actually employ such a system in the real world (e.g. journalists) and evaluation is hence often performed upon general users who may undertake different search patterns to browse through image collections.

### Annotation Task

Rodden and Wood [62] observed that users rarely manually annotate each individual image in a collection (or indeed do not annotate any images at all). One obvious reason for this is the amount of effort required for annotation. Systems such as ImageGrouper [48] and EGO [81] have hence been devised in order to simplify and speed up the annotation of images in a database. Nguyen and Worring [49] run a simulated user study, measuring the number of total interactions required by a user in order to annotate the entire database. They used this method to evaluate a mapping-based visualisation. The baseline number of annotations used is that of a standard linear visualisation which equals the number of images in the database (i.e. one interaction per image). It was shown that the mapping-based visualisation can reduce the number of interactions needed for annotation by up to 94% (dependent on the categories of images in the database and the features used to define similarity). Such a test would obviously be simple to implement in practice, asking users to annotate each image in the database with a keyword from a preset list while measuring the number of interactions or the time spent on the task. In addition, if a ground truth of correct annotations is available, the error rate can also be measured.

### Clustering Study

A novel way of measuring the quality of image clustering is presented by Platt [56]. Two users each used their own personal collections (one of which had corrupt time stamps), and were asked to manually cluster the images into albums which acted as the ground truth for database. Each of the personal collections were then automatically clustered by either time, content, or a combination of the two (as well as a control of equally sized clusters). Each image in the database was used as a query, and based on that the automatic clusterings were compared with the ground-truth (i.e. user-based clusterings), using the number of true positives, false positives and false negatives. These were averaged for all the images in the database to generate a percentage known as the $F1$ metric. It was found that a combination of time and content-based clustering achieved the highest $F1$ score for both the corrupt and non-corrupt image collections.

This approach provides an interesting measurement of the quality of automatic clustering algorithms since it directly compares the results of automatic techniques with those derived manually by a user. The drawback is of course, the time involved to generate the ground truth clustering. In [56] the image collections consisted of 294 and 405 images respectively, whereas for collections of 1,000s of images the task will become not only infeasible, but also prone to human error.

#### User Opinion

After a user study has been conducted, researchers will generally issue the users with a questionnaire in order to gauge their opinions on the different aspects of the system and its user interface. The results of these questionnaires can then be used to modify the system as was done by Rodden *et al.* where the general dislike of users towards image overlapping in MDS visualisations caused the authors to consider fitting the images to a more regular grid structure. When this type of user questioning is included with some other task such as those listed above, it allows to gain an impression on how such a system could be applied in the real world. However, if used without a test such as those listed above, the lack of quantitative statistical data prevents drawing full conclusions upon the true quality of the approach.

### 4.2 Key Findings from User Studies

User evaluations attempt to prove that the system proposed by the authors improves upon methodologies currently used in the field. Sometimes these studies provide interesting insights into how general users gauge these novel browsing systems and additional functions. Perhaps the most significant user studies have been conducted by Rodden *et al.* In [59], a user study was conducted using target search on a randomly assorted grid of images and an MDS visualisation based on image similarity. The authors were able to show:

- Image retrieval is faster when images are arranged by their mutual similarity.
- Users prefer visualisations that do not overlap.
- More distinct images (i.e. images that are on average less similar to all other images in the database) are easier to find.
- Images located closer to the centre of the screen are retrieved faster than those located closer to the edge.

In a later work of Rodden *et al.* [61], an MDS visualisation fitted to a grid is compared first to a system organising images in groups through keywords, and then with a grid arrangement whereby the images are randomly sorted. The users were asked to perform a journalistic task. The key findings from this work were:

- Users prefer the MDS grid visualisation to one arranged through keywords.
- Users are slower at selecting preferred images within the MDS grid visualisation than the randomly assorted grid.

The authors were surprised that users took longer to select images for a travel article using the MDS grid visualisation rather than the randomly assorted grid of images. As a possible reason, they argue that when images are arranged randomly, images appear to be more distinct as it is unlikely that it will be similar to all of their neighbours. However, the authors state that judging from post-test questionnaires, it appeared that users were generally more satisfied with their image selections when using the MDS based interface. This may be because they have selected an image they were looking for in particular, rather than settling for a related image found quickly using the random arrangement.

Rodden and Wood [62] also explored how users manage their digital photographs. Subjects were supplied with a digital camera and a system called Shoebox, an image browsing system arranging images in folders according to the time they were created. Shoebox also has the added functionality of a QBE search facility and a voice annotation system. Findings from this work include:

- The general user has unrealistic expectations of a QBE system, and can find it difficult to improve a query.
- Users are fairly reluctant to manually annotate images, even when provided with a voice annotated system. Only a small percentage of users in the study changed the title of any image in the system.
- Sorting images according to the time at which they were taken allows users to browse the collection by recalling which particular event the image required is from.
- Displaying many image thumbnails at a time decreases the time required for image retrieval.

The work of Rodden *et al.* has looked at arranging images by similarity as well as time; one of the fundamental conclusions from [62] is that displaying as many images as possible to the user improves retrieval time. However an issue with displaying too many image thumbnails is that the user needs to be able to comprehend what is actually depicted within the image.

A zoom facility as described in Section 3.1, allows thumbnails to be displayed within an overview at a fairly low resolution and zoomed into at the user's discretion. A study by Combs and Bederson [9] investigates solely the effect zooming has on improving the user's browsing experience. A Zoomable Image Browser (ZIB) is compared with a traditional image browsing system, whereby image folders can be selected from the left hand pane, and the contents of the folder are displayed in the right hand pane. Enlargement of images can only be performed by opening a new window. The ZIB system provides a keyword search in a top pane, while the results of the search are displayed in a pane located below. The user has the facility to zoom into the search query results. Despite showing that a target search test was faster using ZIB, this was shown not to be statistically significant. The authors of the study also comment that of the 30 users tested, only 50% actually invoked the zoom facility. Combs and Bederson suggest that the number of query results shown at any time was not enough to warrant a zoom facility, as images were displayed at a resolution distinguishable without the zoom requirement. They conclude that a study into the maximum number of images that can be displayed without zoom should be investigated in future work.

Interesting user studies have also been conducted by the developers of the RF browsing systems EGO [81] and ImageGrouper [48]. Both systems are relatively similar, allowing a user to first query the database, then group the images presented in the results in order to modify the feature weights of the internal similarity measure. A target search was performed, asking the user to select ten semantically relevant images to the target. EGO and ImageGrouper were compared with a slider based and a selection based RF system. The key finding from these studies was:

- Image retrieval took longer using the grouping systems rather than the simple relevant or non-relevant selection system.

The authors attributed this to the fact that the drag-and-drop interfaces require more user actions than a simple selection interface.

In clustering-based visualisations, image groups can be displayed to the user in the form of representative images (as discussed in Section 2.2). In [18], the subjective opinion of users was measured for varying forms of the CAT interface. The authors found that:

- Users preferred the CAT interface when representative images were used rather than when representative images were not included.

Unfortunately the evaluation was performed only with 10 users, making it difficult to conclusively state that representative images indeed do improve a user's browsing experience. Future work could use quantitative tests in order to provide a better insight into the effectiveness of representative images as a tool for browsing.

While the CAT interface is an example of a system invoking hierarchical clustering, user studies conducted by Rodden *et al.* were designed to test the effectiveness of dimensionality reduced visualisations. Relatively little work has been conducted into testing which of these different approaches might perform better in terms of image retrieval. Liu *et al.* [42] however do offer some comparison between a clustered and dimensionality reduced approach. The authors were interested to discover how the results from a web image search engine query can be visualised in order to improve the user's browsing experience. Nine users participated using the standard ranked list interface provided by Google image search, an MDS visualisation that could be manually adjusted to fit images to a grid (as described in Section 2.1) and a clustering-based approach which creates five groups of images based on content. Clusters are represented in a left hand pane through a set of the four most representative images in that group. Should the user select the cluster preview thumbnail, all the images from that cluster are displayed in the right hand pane of the browser. 17 queries were performed in which the top 200 images were displayed on each of the interfaces. Users were instructed to browse the results in order to find the images they deemed most relevant to the query terms. Liu *et al.* found that:

- Both the MDS and the clustering-based visualisations clearly outperform the standard ranked list results.
- Although search times were similar between the MDS and cluster visualisations, users clearly preferred the layout of MDS plots characterising it as *"more intuitive and interesting, also convenient for comparing similar images"*.

### 4.3 Discussion

Evaluation of image database navigation systems, more often than not, tends to compare the newly proposed browsing system with a more traditional approach. Various studies have confirmed that image databases visualised, as described earlier in this chapter, do indeed allow for faster retrieval than traditional linear approaches [19, 42, 59].

What the majority of user studies have not been able to show is how their browsing system can perform against other browsing systems discussed in this chapter. For example, little work exists in comparing a hierarchically clustered visualisation of a database against the same database visualised using MDS. While the study of Liu *et al.* [42] does offer such a comparison based on a target search scenario, the results were too close to conclude which of the two paradigms offers a more efficient way of searching.

Analysis of questionnaires returned by users after testing allows the collection of personal preferences. However, unlike e.g. the time required to perform a test which can be statistically analysed, user opinion is highly subjective and is often linked to the background and environment of the test population. In addition, the majority of user studies conducted are based on a relatively small number of participants, often no more than ten. Clearly, drawing statistically relevant conclusions from such a small sample size is difficult, both for quantitive measures such as search time and for subjective opinions collected through questionnaires (including those where subjects are asked to assign scores on an ordinal scale).

Future work should focus on developing a standardised benchmark that could be used within the browsing community in order to fully gauge the quality of a newly developed system. This benchmark could comprise both specific search tasks (such as target search) and annotation tasks (similar to the one adopted in [49]) which can be applied to any image database navigation system. Such a benchmark would of course require an underlying dataset with a ground truth (e.g. manual annotations) which in itself is not straightforward to obtain due to the work involved and other factors such as copyright issues.

## 5 Conclusions

In this chapter we have investigated, in detail, the current state-of-the-art of image retrieval systems that allow a user to visually navigate through an image collection. We first looked at similarity-based methods providing an intuitive visualisation of an image collection and identified three main approaches. Mapping-based visualisations maintain the relationships between images in the database in the high-dimensional feature space. Projection into the (typically 2-dimensional) visualisation space is achieved through application of dimensionality reduction techniques such as PCA or MDS. This type of visualisation has also been adopted in systems that employ virtual reality concepts to provide a more immersive browsing experience. However, the costs associated with the necessary equipment will prevent wide-spread adoption of this approach. Clustering-based methods employ, as the name implies, a clustering algorithm to organise images in a collection. Clustering the images into smaller groups of similar images allows the user to browse down a hierarchy, whereby the further down the tree they delve, the more similar images become. Graph-based visualisations express relationships between images (such as visual similarity or common keyword annotation) as links of a graph structure that is visualised to the user. Image collections can also be displayed based on time stamp information which can prove useful to identify distinct events and display relevant pictures.

Mapping-based visualisations aim to maintain the relationships between images occurring in the high-dimensional feature space, and display them usually within the 2D constraints of a computer display or a 3D virtual environment. It has been shown in [59] that arranging images according to visual similarity can reduce the time required for image retrieval. These visualisations harness the power of the human cognitive system, passing a vast quantity of data processing subconsciously to the user's mind. However, one of the drawbacks of this type of visualisation is that the limited space often causes images to overlap or to occlude each other. Ways to address this issue and reduce overlap include the fitting of images to a regular grid structure or slight adjustments of the visual arrangement in order to preserve the structure. Another problem with mapping-based visualisations is that they are computationally expensive to generate, and are hence rarely suitable for computing 'on-the-fly' visualisations of a large number of query results. Furthermore, the addition of images to the collection typically requires these visualisations to be recalculated in their entirety.

Clustering-based visualisations have the advantage that by dividing the database into smaller entities, only a small subset of images needs to be visualised. This ultimately leads to less processing for both the system and the user. The system needs only to load a section of the database when the user has accessed a particular cluster of images, rather than loading all images as is the case with global mapping-based visualisations. The cognitive load on the user is also reduced as the number of distinct images to be inspected is much lower. However, a disadvantage of clustered visualisations is that the user can become 'trapped' in a subset of the database. This can occur when representative images used at higher levels of the tree either do not represent the images in that subtree well enough, or are not distinct enough from other representative images at the same level. Both scenarios can lead to the user traversing nodes of the structure in vain, leading to excessive time required for retrieval and added frustration to the user. It should be noted that this is not so much a flaw of the visualisation itself but is rather caused by the underlying similarity measure employed, the best of which are still incapable of modeling human perception appropriately.

Indeed, this problem applies to all forms of visualisations including graph-based approaches. If the features extracted and similarity measures do not model human perception well, the links formed between images may impede rather than support the browsing experience. Creating links to multiple images based on a variety of features, as implemented in $\mathrm{NN}^k$ networks [22], allows the user to browse through images in the database based upon a particular feature such as colour or texture. However, the user may need to adjust the number of neighbours displayed as an excessive number of links between images will make the visualisation more complex and less intuitive.

The variety of browsing tools available to the user are usually common to all visualisations. Being able to zoom into areas of interest can be applied to any visualisation, although vertical zooming is available only in hierarchically organised visualisations. These structures often come with some overview of the underlying tree and the user's current location within the system in order to help with the navigation task. This kind of overview can also be applied to mapping-based and graph-based visualisations, particularly when the user has increased the zoom factor so that not all of the visualisation is visible within the screen area. In this case a panning function is required to allow the user to navigate the structure without having to zoom out again. Manual scaling of images and magnified previews of images can further enhance the user's browsing experience.

Relatively little work has been performed into investigating which visualisation paradigm may be the most useful, although a variety of user studies have shown that organising image databases in the ways presented in this chapter can reduce the retrieval time when compared with traditional approaches. What is currently missing is a standard benchmark for assessing the effectiveness and efficiency of browsing systems. Such a benchmark would be based on a standardised image set (or several collections of different magnitudes in order to be able to judge scalability) together with a ground truth and a number of pre-defined tasks. One such task which seems particularly interesting is the annotation task defined in [49], a 'real world' task which can be quantitatively measured in order to compare systems. A large, copyright-free image database however is still an issue, although systems such as those presented in [13, 39] use the online image resource Flickr [17] to obtain images. Defining a ground truth is an even bigger challenge (as can e.g. be seen immediately by inspecting the annotations that are given on Flickr).

In addition to advancements in the evaluation of image database navigation systems, further research and new browsing paradigms are likely to be required to harness the true potential of image browsing. One of the coming challenges for browsing systems is the decreasing screen resolution and reduced processing available, that have come as a consequence of mobile computing. More and more people use their mobile phones to explore the internet, and require access to the millions of images available online. Nowadays, mobile phones also act as a primary source of image capture for many. Photographs are often uploaded to the web to either share on social networking sites, or uploaded to a 'cloud' (or server) whereby the user can access their images from any device. Works such as [21, 70, 84] have looked at developing traditional QBE CBIR systems for mobile devices, whilst [32] does briefly look at browsing on a mobile device. With the increasing graphical and processing ability of handheld devices, coupled with the increasing number of images stored locally and online, browsing large image databases in the palm of the users hand will almost certainly be a future requirement.

## References

1. Abdel-Mottaleb, M., Krischnamachari, S., Mankovich, N.J.: Performance Evaluation of Clustering Algorithms for Scalable Image Retrieval. In: IEEE Computer Society Workshop on Empirical Evaluation of Computer Vision Algorithms (1998)
2. Assfalg, J., Del-Bimbo, A., Pala, P.: Virtual Reality for Image Retrieval. Journal of Visual Languages and Computing 11(2), 105–124 (2000)
3. Atkinson, K.E.: An Introduction to Numerical Analysis. John Wiley and Sons, Chichester (1989)
4. Bederson, B.: Quantum Treemaps and Bubblemaps for a Zoomable Image Browser. In: ACM Symposium on User Interface Software and Technology, pp. 71–80 (2001)
5. Borth, D., Schulze, C., Ulges, A., Breuel, T.: Navidgator - Similarity Based Browsing for Image and Video Databases. In: German Conference on Advances in Artificial Intelligence, pp. 22–29 (2008)
6. Chen, C.: Information Visualization. Springer, Heidelberg (2004)

7. Chen, C., Gagaudakis, G., Rosin, P.: Similarity-Based Image Browsing. In: International Conference on Intelligent Information Processing, pp. 206–213 (2000)
8. Chen, Y., Butz, A.: Photosim: Tightly Integrating Image Analysis into a Photo Browsing UI. In: International Symposium on Smart Graphics (2008)
9. Combs, T., Bederson, B.: Does Zooming Improve Image Browsing? In: ACM Conference on Digital Libraries, pp. 130–137 (1999)
10. Cruz-Neira, C., Sandin, D., DeFanti, T.: Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE. In: 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 135–142 (1993)
11. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image Retrieval: Ideas, Influences, and Trends of the New Age. ACM Computing Surveys 40(2), 1–60 (2008)
12. Deng, D., Zhang, J., Purvis, M.: Visualisation and Comparison of Image Collections based on Self-organised Maps. In: Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation, pp. 97–102 (2004)
13. Dontcheva, M., Agrawala, M., Cohen, M.: Metadata Visualization for Image Browsing. In: ACM Symposium on User Interface Software and Technology (2005)
14. Eidenberger, H.: A Video Browsing Application Based on Visual MPEG-7 Descriptors and Self-organising Maps. International Journal of Fuzzy Systems 6(3) (2004)
15. Faloutsos, C., Equitz, W., Flickner, M., Niblack, W., Petkovic, D., Barber, R.: Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems 3, 231–262 (1994)
16. Faloutsos, C., Lin, K.: FastMap: A Fast Algorithm for Indexing, Datamining and Visualization of Traditional and Multimedia Datasets. In: ACM SIGMOD International Conference on Management of Data, pp. 163–174 (1995)
17. Flickr (2009), `http://www.flickr.com/`
18. Gomi, A., Miyazaki, R., Itoh, T., Li, J.: CAT: A Hierarchical Image Browser Using a Rectangle Packing Technique. In: International Conference on Information Visualization, pp. 82–87 (2008)
19. Graham, A., Garcia-Molina, H., Paepcke, A., Winograd, T.: Time as Essence for Photo Browsing through Personal Digital Libraries. In: ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 326–335 (2002)
20. Gupta, A., Jain, R.: Visual Information Retrieval. Communications of the ACM 40(5), 70–79 (1997)
21. Hare, J.S., Lewis, P.H.: Content-based Image Retrieval Using a Mobile Device as a Novel Interface. In: SPIE Storage and Retrieval Methods and Applications for Multimedia, pp. 64–75 (2005)
22. Heesch, D., Rüger, S.M.: $\mathrm{NN}^k$ Networks for Content-Based Image Retrieval. In: McDonald, S., Tait, J.I. (eds.) ECIR 2004. LNCS, vol. 2997, pp. 253–266. Springer, Heidelberg (2004)
23. Heesch, D., Rüger, S.: Three Interfaces for Content-Based Access to Image Collections. In: International Conference on Image and Video Retrieval, pp. 491–499 (2004)
24. Hilliges, O., Baur, D., Butz, A.: Photohelix: Browsing, Sorting and Sharing Digital Photo Collections. In: IEEE Tabletop Workshop on Horizontal Interactive Human-Computer Systems, pp. 87–94 (2007)

25. Hilliges, O., Kunath, P., Pryakhin, A., Butz, A., Kriegel, H.P.: Browsing and Sorting Digital Pictures using Automatic Image Classification and Quality Analysis. In: International Conference on Human-Computer Interaction, pp. 882–891 (2007)
26. Hinton, G., Roweis, S.: Stochastic Neighbor Embedding. In: Advances in Neural Information Processing Systems, vol. 15, pp. 833–840 (2002)
27. Apple iPhoto (2009), `http://www.apple.com/ilife/iphoto/`
28. Jacobs, C.E., Finkelstein, A., Salesin, D.H.: Fast Multiresolution Image Querying. In: Conference on Computer Graphics and Interactive Techniques, pp. 277–286 (1995)
29. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
30. Jambu, M.: Exploratory and Multivariate Data Analysis. Academic Press, London (1991)
31. Keller, I., Meiers, T., Ellerbrock, T., Sikora, T.: Image Browsing with PCA-Assisted User-Interaction. In: IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 102–108 (2001)
32. Khella, A., Bederson, B.: Pocket PhotoMesa: A Zooming Image Browser for PDA's. In: International Conference on Mobile and Ubiquitous Multimedia, pp. 19–24 (2004)
33. Kohonen, T.: Self-organizing Maps. Springer, Heidelberg (1997)
34. Koikkalainen, P., Oja, E.: Self-organizing Hierarchical Feature Maps. In: International Joint Conference on Neural Networks, vol. 2, pp. 279–285 (1990)
35. Krischnamachari, S., Abdel-Mottaleb, M.: Image Browsing using Hierarchical Clustering. In: IEEE Symposium Computers and Communications, pp. 301–307 (1999)
36. Kruskal, J.B., Wish, M.: Multidimensional Scaling. Sage, Thousand Oaks (1978)
37. Laaksonen, J., Koskela, M., Oja, E.: PicSOM – Self-organizing Image Retrieval with MPEG-7 Content Descriptors. IEEE Transactions on Neural Networks: Special Issue on Multimedia Processing 13(4), 841–853 (2002)
38. Lew, M.S., Sebe, N.: Visual Websearching Using Iconic Queries. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 788–789 (2000)
39. Li, J., Wang, J.Z.: Real-Time Computerized Annotation of Pictures. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(6), 985–1002 (2008)
40. Lim, S., Chen, L., Lu, G., Smith, R.: Browsing Texture Image Databases. In: International Conference on Multimedia Modelling, pp. 328–333 (2005)
41. Linde, Y., Buzo, A., Gray, R.: An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications 28, 84–94 (1980)
42. Liu, H., Xie, X., Tang, X., Li, Z.W., Ma, W.Y.: Effective Browsing of Web Image Search Results. In: ACM International Workshop on Multimedia Information Retrieval, pp. 84–90 (2004)
43. Ma, W.Y., Manjunath, B.S.: NeTra: A Toolbox for Navigating Large Image Databases. Multimedia Systems 7(3), 184–198 (1999)
44. Milanese, R., Squire, D., Pun, T.: Correspondence Analysis and Hierarchical Indexing for Content-Based Image Retrieval. In: IEEE International Conference on Image Processing, pp. 859–862 (1996)
45. Moghaddam, B., Tian, Q., Lesh, N., Shen, C., Huang, T.: Visualization and User-Modeling for Browsing Personal Photo Libraries. International Journal of Computer Vision 56(1/2), 109–130 (2004)

46. Müller, H., Müller, W., Squire, D.M., Marchand-Maillet, S., Pun, T.: Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals. Pattern Recognition Letters 22(5), 593–601 (2001)
47. Nakazato, M., Huang, T.: 3D MARS: Immersive Virtual Reality for Content-Based Image Retrieval. In: IEEE International Conference on Multimedia and Expo., pp. 44–47 (2001)
48. Nakazato, M., Manola, L., Huang, T.: ImageGrouper: A Group-Oriented User Interface for Content-Based Image Retrieval and Digital Image Arrangement. Journal of Visual Language and Computing 14(4), 363–386 (2003)
49. Nguyen, G.P., Worring, M.: Interactive Access to Large Image Collections using Similarity Based Visualization. Journal of Visual Languages and Computing 19, 203–224 (2008)
50. Osman, T., Thakker, D., Schaefer, G., Lakin, P.: An Integrative Semantic Framework for Image Annotation and Retrieval. In: IEEE/WIC/ACM International Conference on Web Intelligence, pp. 366–373 (2007)
51. Pecenovic, Z., Do, M.N., Vetterli, M., Pu, P.: Integrated Browsing and Searching of Large Image Collections. In: Laurini, R. (ed.) VISUAL 2000. LNCS, vol. 1929, pp. 279–289. Springer, Heidelberg (2000)
52. Pentland, A., Picard, W.R., Sclaroff, S.: Photobook: Content-Based Manipulation of Image Databases. International Journal of Computer Vision 18(3), 233–254 (1996)
53. Google Picasa (2009), `http://picasa.google.com/`
54. Plaisant, C., Carr, D., Shneiderman, B.: Image Browsers: Taxonomy, Guidelines, and Informal Specifications. IEEE Software 12, 21–32 (1995)
55. Platt, J., Czerwinski, M., Field, B.: PhotoTOC: Automatic Clustering for Browsing Personal Photographs. Technical report, Microsoft Research (2002)
56. Platt, J.C.: AutoAlbum: Clustering Digital Photographs using Probalistic Model Merging. In: IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 96–100 (2000)
57. Porta, M.: New Visualization Modes for Effective Image Presentation. International Journal of Image and Graphics 9(1), 27–49 (2009)
58. Rodden, K.: Evaluating Similarity-Based Visualisations as Interfaces for Image Browsing. PhD thesis, University of Cambridge Computer Laboratory (2001)
59. Rodden, K., Basalaj, W., Sinclair, D., Wood, K.: Evaluating a Visualisation of Image Similarity as a Tool for Image Browsing. In: IEEE Symposium on Information Visualisation, pp. 36–43 (1999)
60. Rodden, K., Basalaj, W., Sinclair, D., Wood, K.: A Comparison of Measures for Visualising Image Similarity. In: The Challenge of Image Retrieval (2000)
61. Rodden, K., Basalaj, W., Sinclair, D., Wood, K.: Does Organisation by Similarity Assist Image Browsing? In: SIGCHI Conference on Human Factors in Computing Systems, pp. 190–197 (2001)
62. Rodden, K., Wood, K.: How Do People Manage Their Digital Photographs? In: SIGCHI Conference on Human Factors in Computing Systems, pp. 409–416 (2003)
63. Roweis, S., Saul, L.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290(5500), 2323–2326 (2000)
64. Rubner, Y., Guibas, L.J., Tomasi, C.: The Earth Movers Distance, Multidimensional Scaling, and Color-based Image Retrieval. In: APRA Image Understanding Workshop, pp. 661–668 (1997)

65. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, M.: Relevance Feedback: A Power Tool for Interactive Content-based Image Retrieval. IEEE Transaction on Circuits and Systems for Video Technology 8(5), 644–655 (1998)
66. Ruszala, S., Schaefer, G.: Visualisation Models for Image Databases: A Comparison of Six Approaches. In: Irish Machine Vision and Image Processing Conference, pp. 186–191 (2004)
67. Sammon, J.W.: A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers 18(5), 401–409 (1969)
68. Santini, S., Jain, R.: Integrated Browsing and Querying for Image Databases. IEEE Multimedia 7, 26–39 (2000)
69. Sarkar, M., Brown, M.: Graphical Fisheye Views. Communications of the ACM 37(12), 73–83 (1994)
70. Sarvas, R., Herrarte, E., Wilhelm, A., Davis, M.: Metadata Creation System for Mobile Images. In: International Conference on Mobile Systems, Applications, and Services, pp. 36–48 (2004)
71. Schaefer, G., Ruszala, S.: Image database navigation: A globe-al approach. In: Bebis, G., Boyle, R., Koracin, D., Parvin, B. (eds.) ISVC 2005. LNCS, vol. 3804, pp. 279–286. Springer, Heidelberg (2005)
72. Schaefer, G., Ruszala, S.: Image Database Navigation on a Hierarchical Hue Sphere. In: International Symposium on Visual Computing, pp. 814–823 (2006)
73. Schaefer, G., Ruszala, S.: Image Database Navigation on a Hierarchical MDS Grid. In: 28th Pattern Recognition Symposium, pp. 304–313 (2006)
74. Schaefer, G., Stich, M.: UCID – An Uncompressed Colour Image Database. In: Storage and Retrieval Methods and Applications for Multimedia, pp. 472–480 (2004)
75. Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-Based Image Retrieval at the End of the Early Years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1349–1380 (2000)
76. Swain, M., Ballard, D.: Color Indexing. International Journal of Computer Vision 7(1), 11–32 (1991)
77. Tao, D., Tang, X., Li, X., Rui, Y.: Direct Kernal Biased Discriminant Analysis: A New Content-Based Image Retrieval Relevance Feedback Algorithm. IEEE Transactions on Multimedia 8(4), 716–727 (2006)
78. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric Bagging and Random Subspace for Support Vector Machines-Based Relevance Feedback in Image Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(7), 1088–1099 (2006)
79. Tenenbaum, J., Silva, V., Langford, J.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290(5500), 2319–2322 (2000)
80. Tian, G.Y., Taylor, D.: Colour Image Retrieval Using Virtual Reality. In: IEEE International Conference on Information Visualization, pp. 221–225 (2000)
81. Urban, J., Jose, J.M.: EGO: A Personalized Multimedia Management and Retrieval Tool. International Journal of Intelligent Systems 21(7), 725–745 (2006)
82. van Liere, R., de Leeuw, W.: Exploration of Large Image Collections Using Virtual Reality Devices. In: Workshop on New Paradigms in Information Visualization and Manipulation, held in conjunction with the 8th ACM International Conference on Information and Knowledge Management, pp. 83–86 (1999)
83. Worring, M., de Rooij, O., van Rijn, T.: Browsing Visual Collections Using Graphs. In: International Workshop on Multimedia Information Retrieval, pp. 307–312 (2007)

84. Yeh, T., Tollmar, K., Darrell, T.: Searching the Web with Mobile Images for Location Recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 76–81 (2004)
85. Zhang, H., Zhong, D.: A Scheme for Visual Feature Based Image Indexing. In: SPIE/IS&T Conference on Storage and Retrieval for Image and Video Databases, pp. 36–46 (1995)
86. Zhou, X., Huang, T.: Relevance Feedback in Image Retrieval: A Comprehensive Review. Multimedia Systems 8(6), 536–544 (2003)