Urban, J. and Jose, J.M. (2006) EGO: a personalised multimedia management tool. *International Journal of Intelligent Systems 21*(7):pp. 725-745.

# EGO: A Personalised Multimedia Management Tool

Jana Urban and Joemon M. Jose

Department of Computing Science, University of Glasgow, Glasgow G12 8RZ, UK
{jana,jj}@dcs.gla.ac.uk

**Abstract.** The problems of Content-Based Image Retrieval (CBIR) systems can be attributed to the *semantic gap* between the low-level data representation and the high-level concepts the user associates with images, on the one hand, and the time-varying and often vague nature of the underlying information need, on the other. These problems can be addressed by improving the interaction between the user and the system. In this paper, we sketch the development of CBIR interfaces, and introduce our view on how to solve some of the problems of the studied interfaces. To address the semantic gap and long-term multifaceted information needs, we propose a *"retrieval in context"* system. EGO is a tool for the management of image collections, supporting the user through personalisation and adaptation. We will describe how it learns from the user's personal organisation, allowing it to recommend relevant images to the user. The recommendation algorithm is detailed, which is based on relevance feedback techniques.

## 1 Introduction

Most research in the field of Information Retrieval (IR) focuses on the performance of the retrieval system. In an effort to broaden the horizon of future search systems, researchers attribute increasingly more importance to the human-computer interaction side of IR. Based on studies of information-seeking behaviour[1, 2], and user emotions and psychologies [3], new interfaces for search systems move into focus. Belkin has recently pointed out some grand challenges for information system design [4]. By asking the question *"What might be the next steps to take in system design to support information seeking?"*, he identifies two issues, namely:

1. To design a system that supports a variety of interactions, and
2. Personalising the support of information interaction.

In this line, the aim of our work is to design an adaptive Content-based Image Retrieval (CBIR) system. The above issues act as the primary design goals for its development. We argue the system should be flexible, adapting to the diversity of users by supporting a variety of interactions. Especially, interactions should not be limited to strict image retrieval, but the retrieval aspect should be placed in a wider context of the work environment. What is sought for is a holistic application for retrieval, organisation, annotation, and whatever other activity a user might wish to do. Moreover, the system should have the ability to adapt to the user by learning from user interaction. This facilitates personalisation and improves system performance, since the system is able to learn from the knowledge and interaction of individual users.

In this paper we introduce our system, EGO (Effective Group Organisation), which is currently being developed with these principles in mind. The system addresses many of the shortcomings of traditional CBIR systems, which we identify and discuss in the following sections. In Section 2 we provide our motivations for researching novel interaction strategies. Section 3 reviews a representative selection of CBIR systems to trace their development. The remaining sections are devoted to the EGO system. Section 4 provides a detailed description of its interface and some of the implementation details, followed by a list of features we are still working on and a concluding discussion in Sections 5 and 6, respectively.

## 2 Motivation

CBIR is an active research area sharing the effort of making the vast amount of available images more manageable, and one can observe a constant progress towards more and more intelligent systems [5]. The benefits this provides are overshadowed by problems caused by the interaction with a typical CBIR user interface. The most persistent questions are:

– *"What is the meaning of an image?"*

  It can be difficult to grasp the *meaning* of an image. As a work of art—similar to poems—an image's meaning cannot be pinpointed with universal consensus. Current CBIR technology has difficulties in extracting the major objects contained in an image, let alone its meaning. The trend to overcome the *semantic gap* between the system and the user, is to learn semantic concepts in order to move closer to decoding meaning [6, 7]. Since there is no consensus on universal meaning, the semantic concepts depicted in, or otherwise emerging from, an image are individual to a user. The dependency of semantic concepts on individual *interpretation* and *context* has been widely acknowledged in the CBIR literature [7, 8].

– *"How can the user be assisted in communicating their information need?"*

  The query formulation problem has emerged as an IR problem in general [9]. The internal representation of documents is optimised for indexing efficiency and retrieval performance, but most often than not rather alien to the user. Hence, users may be taught to speak the language of the system, or even better the system taught to speak the language of its users.

  Relevance feedback (RF) is an automatic process of improving the initial query based on relevance judgements provided by the user [10–13]. The process is aimed at relieving the user from having to reformulate the query in order to improve the retrieval results incrementally. The search becomes more intuitive to the user, since they are only requested to label the returned images as either relevant or not. However, it is still an ongoing research challenge to accurately learn the information need from the user based on a few relevance judgements [14].

– *"How can the time-varying nature of information needs be modelled in the system?"*

  It is often the case that the initial idea of an image the user has in mind before starting a search session deviates from the final results they will choose [15]. Whatever the reasons for this change, it shows that it is hard to assume to be able to guess an ideal query from the initial query and consequent relevance feedback. Rather it should be attempted to trace the actions over the iterations in order to detect changes in the information need [15, 16].

  Although there are some approaches that model the dynamic nature of information needs [15, 17], still more work needs to be done to this end. Current RF techniques treat the relevance judgements gained over a number of iterations homogeneously, sometimes even collecting them all in a pool before starting the learning procedure. It would be more beneficial if the relevance judgements were *traced* rather than *collected*. In this way, new feedback can be compared to previous feedback, with the goal to detect changes over iterations.

To be able to find an even moderately satisfactory solution to any of these questions, it has become apparent that the *user* plays a very—if not *the* most—important role. Without the users' knowledge of the world and their superior visual system, CBIR system capabilities are rather limited. Moreover, user satisfaction greatly depends on subjective judgements of image contents and relevance. It is impossible to accommodate the huge diversity of users, yet systems can adjust to individual users by learning their preferences.

From the user's perspective, however, searching for and performing a selection of images is usually embedded in other tasks, and thus it is at least equally important to understand and capture the work flow [18, 19]. Therefore, a solution to accommodate the needs of users must be flexible, support multiple tasks, and allow exchanges or even seamless integration with other applications used for the work tasks. Moreover, the search process often takes place in a collaborative context, in which people work together, are inspired and learn from each other's activities.

What is needed is a *holistic view* on personal image organisation and retrieval. Satisfactory solutions to the uncertainty of image meaning, the query formulation problem and the time-varying needs of the user, can be found in this environment only. A "retrieval in context system" offers a great opportunity for learning, adaptation and personalisation. We now review the related work that has led us to the design of EGO.

## 3 Related Work

In this section we study existing solutions for user interface support for CBIR. As we will see shortly, existing systems are predominantly search systems. They lack the ability to support the user in organising their results in a meaningful and time-persistent way, thus losing the opportunity of learning and adapting to the user- and task-dependent context.

The interface is the mediator between the user and the search system. From the perspective of the user, it is the entry point to the system. A properly designed interface assists the user with meaningful and intuitive ways of communicating their information need to the system and displays results in ways that stimulate the user and enhance performance. In this section we will introduce approaches for creating more meaningful result displays and review interfaces to see how they deal with the interaction metaphor. The major innovation for the former has been to replace the traditional linear result display, ranked by similarity to the query, with two- or three-dimensional maps of the returned images [20, 21]. These multidimensional displays aim at revealing relationships between images by visualising mutual similarities between any two images. The axes either represent feature dimensions, such as colour or textures, or are a result of dimension reduction methods, such as Principal Component Analysis (PCA), mapping the cardinality of the feature space down to the two or three most discriminative dimensions. The goal of these visualisation techniques is to show the images in their surroundings or context. By depicting relationships between images in a global view, the user can form a more accurate mental model of the database and support navigation within it. A user study conducted by Rodden et al. [22] has pointed to the benefits of a display organised by similarity for image browsing. The remainder of this section serves as an outline of the development of CBIR systems on the basis of their interface design.

**QBIC** is one of the earliest image retrieval systems with CBIR query facilities, developed by IBM [23]. QBIC supports the retrieval of images based on a number of primitive image features, including colour, texture, and shape. The query component is the most important aspect of the interface. It allows the user to compose a query by drawing the rough shapes and choosing the colour of objects according to the spatial layout the objects in the retrieved image should convey. The query specified in this manner is automatically translated into the primitive features used for indexing the database images. After issuing the query to the system, the resulting images are displayed in a grid sorted by decreasing similarity scores to the query features. This interface requires intuitive and meaningful query composition facilities, and is reliant on the user's ability to map from the high-level concepts they have in mind when querying to the low-level visual attributes the system understands for retrieval. It hardly assists the user in this task, and does not learn from user interaction.

**MARS:** To alleviate the user from the query formulation problem, recent systems have emphasised an interactive result refinement strategy made possible through relevance feedback. To initiate the search, these systems usually implement the *"Query-by-example"* approach. There one user-supplied image from which the query features are extracted is used to bootstrap the search. After the first iteration, the user is asked to specify the relevance of images in the result set. In MARS [12] this feedback can be given by changing the value of a slider of any image indicating the degree of relevance when pushed to one side, or irrelevance when pushed to the other. The system responds with a new result set, which could be improved through a suitable learning strategy from the experience gained from the relevance feedback. This process is iterated until the user is satisfied with the results. Hence, interaction takes place between the system and the user, in which the user responds to the result set of images returned by the system, and the system responds to the relevance feedback given by the user. The requirements for the interface are minimal in this case. Apart from letting the user choose an initial query image (or alternatively start with a random set of images), the user must be able to associate some relevance values for any of the images in the result set. Nevertheless, the system does not provide sufficient information to assist the user in making vital decisions. For instance, the system does not give any indication of how many images to select for feedback, which images to select, what kind of effect a selected image has on the new results, etc. As a result, the user is forced to make decisions without having enough knowledge about the effects of their actions. Since the actions are usually irreversible this can have detrimental effects on the perceived performance of the system.

**ImageGrouper** presents novel interaction strategies for CBIR [13]. The major emphasis lies on group-based search, and this system combines the tasks of searching, annotating, and organising images by groups. Image retrieval in this interface follows the *trial-and-error* approach as opposed to the traditional *incremental* search of most CBIR systems that incorporate relevance feedback. It is supported by separating the results display from the workspace. The workspace serves as the organisation and storage area. Images can easily be dragged from the results panel onto the workspace, and consequently be organised into groups. Groups are created by drawing a rectangle around a cluster of images. For relevance feedback, the groups can be classified as positive, negative or neutral groups. The introduction of a separate workspace ensures that all images used for relevance feedback and their organisation are always visible. By dragging images around the workspace, ie in and out of groups, and selecting different groups as negative or positive examples, a *trial-and-error* search is easily supported. This relies on lightweight operations of creating groups (draw rectangle), assigning images to groups (drag'n'drop), and labelling the groups (simple popup menu). Through a simple interaction strategy the user gives relevance feedback information without having to think in terms of the system's internal representation. The organisation of images into groups is more natural to the user and matches more closely to the process of accomplishing the task. The *trial-and-error* approach ensures that actions are reversible, which is necessary due to the inferior capabilities of current CBIR technology of matching human similarity judgements. Nonetheless, ImageGrouper fails to deal with varying types of information need. The system learns to improve its retrieval results in order to satisfy the current information need. Although groups can be saved for later use, the contextual information they convey is not used to adapt the system in the long run.

In summary, the representation of information has traditionally been confined to those suitable for retrieval. Thus, in image retrieval systems the interface was focused on the provision of query components to specify the appropriate image features used for retrieval. However, in order to support the way information is used and managed, the interface has to include better result handling and personalisation techniques. ImageGrouper moves some way towards this goal, nevertheless, it is still mainly a *search* interface. Our aim is to develop a tool, EGO, that places emphasis on the long-term management and personalised access to the image (or multimedia) collection. The long-term usage provides additional search clues such as usage histories of images and groups that should be combined with the low-level image features. Similar to the Fetch [24] system for organising web documents, EGO provides the means to describe a long-term multifaceted information need. To achieve this, the user and the system interactively group potentially similar images. The process of grouping images stretches over multiple sessions, so that existing groups are changed and new ones are created whenever the user interacts with the collection. By placing them on a workspace, the user leaves trails of her actions for her or others to inspect and follow. The process is incremental and dynamic: an organisation is built up and changes by usage. A semantic organisation emerges that reflects the user's mental model and the work tasks. These are the two most important influences on the organisation of personal media recognised in [25]: *"There is no unique or right model; rather the mental model is personal, has meaning for the individual who creates it, and is tied to a specific task."* As will be described in the following sections, EGO is a personalised "retrieval in context" system that allows the user to effectively manage and search their images. It captures both short- and long-term information needs, communicated by leaving behind trails of actions, and used by the system to adapt to the user's need.

## 4 EGO: Effective Group Organisation

The main idea that drives the system design is to provide an environment for the day-to-day usage of the data, in which both search and organisation processes take place and are interleaved with each other. In this section we provide the general concepts of the system, describe how it would typically be used and how the system adapts based on the interaction.

The prototype interface for EGO is depicted in Fig. 1. In EGO the user will be involved in an organisation process, in which the user and the system interactively group images. As a starting point, the system provides a query panel (in the top left-hand corner in Fig. 1), in which traditional query-by-example queries can be issued. The search results will be displayed in the panel beside. The user can then drag images from the results into groups on the workspace. This forms the start of the interactive group creation. For the currently selected group the system provides recommendations of new images based on the images

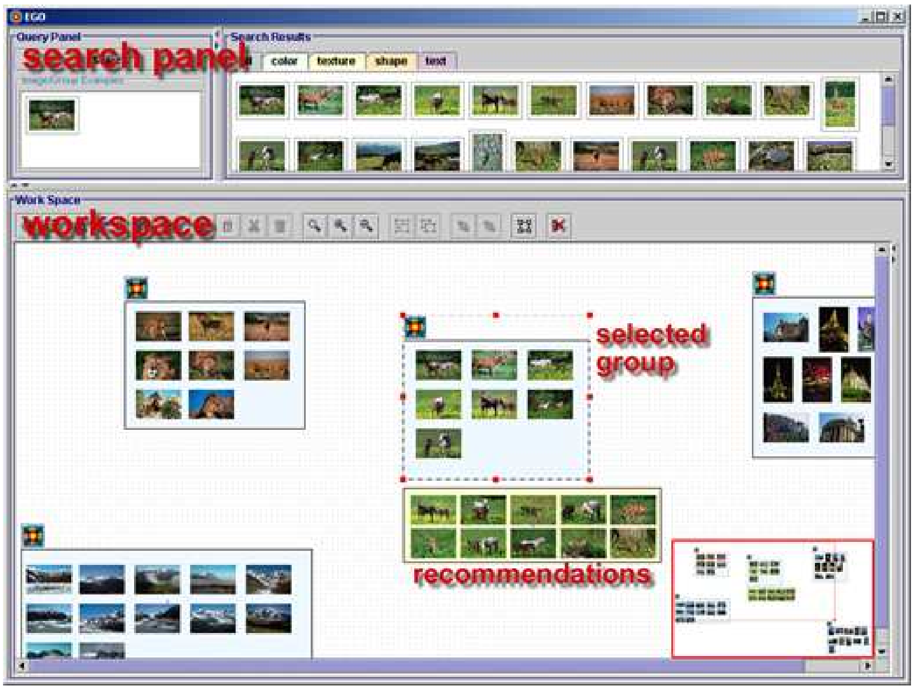



**Fig. 1.** The annotated interface.

already contained in the group at certain time intervals (or on user request). The system's suggestions appear as a popup (yellow rectangle) below the currently selected group. The user can select recommended images to add (by dragging them into the group), and the system will update its previous suggestions. This process can iterate as long as the user is looking for more images to add to that group. We will now look more closely at the components making up the interface.

### 4.1 The Interface

The EGO interface comprises a query editor, results display area and workspace. By providing these facilities, different types of requirements are catered for, enabling the user to both search and organise results effectively. In the following, the main components are discussed in more detail.

**The Search Panel:** The upper half of the screen is devoted to the search facilities. It consists of the query and the results display panel. It should be noted that the size of all main components in the interface can be changed or even hidden on demand. In the query panel, the user can issue a search by choosing example query images. At the moment, the query-by-example strategy is the only one supported in EGO. Having argued for a flexible interface, other querying modes will be provided in the future. This might include keyword search (if annotation is available, see below in Sect. 5), or other query construction facilities based on semantic concepts (*eg*, [26]). The search results are displayed in the results panel beside the query construction component. It allows for different views of the results based on the features supported. The user can choose to view all results or results for only one feature category (colour, texture, shape or text, respectively). Again, we have only implemented a linear result display, but other visualisation techniques, such as the ones mentioned above [20, 21] or that of [15], would be an additional enhancement of the system. The search component provides the user with a basic query facility to search the database, which is useful for both the fulfillment of very specific information needs and serves as an entry point to the collection. From the search results the user can easily drag relevant images onto the workspace to start organising the collection.

**The Workspace:** Similar to the ImageGrouper [13] and Fetch [24] systems, the main component of the interface is the workspace panel provided in EGO. The workspace serves as an organisation ground for the user to construct groupings of images. Groups can be created by right-clicking anywhere on the workspace, which opens a context menu in which the option can be selected. Traditional drag-and-drop techniques allow the user to drag images into a group, reposition the group on the workspace or the images within a group. It has to be noted that, unlike in conventional file systems, an image can belong to multiple groups simultaneously. The workspace is designed as a potentially infinite space to accommodate a large number of groups. Panning and zooming techniques are supported to assist navigation in a large information space. Additionally, a bird's eye view of the workspace is available in the bottom right-hand corner of the interface. It provides an overview, in which the whole workspace is visible, and a sense of location by marking the position of the current view. Additionally, a fish-eye view could be beneficial to provide a view of the whole organisation and reduce clutter.

The act of grouping information is a natural means of managing information to support diverse, complex, and often simultaneous tasks [24]. This metaphor allows the user to resort to traditional problem solving techniques, freeing her from the necessity of query formulation, which should ultimately create a natural and enhanced information-seeking environment. In addition, varying types of information need can be supported. Short-term needs can be satisfied by locating previously created groups that best match a user's need. (While an automatic location of groups based on a query is still in the process of being implemented, the user can still manually locate groups in the workspace.) If there are no matching groups, the user can still resort to the traditional query facility. Furthermore, groups can be created and populated over time, reflecting long-term, time-varying needs.

To assist the user in this task, EGO includes a recommendation system. The recommendation system observes the user's actions, which enables it to adapt to their information requirements and to make suggestions of potentially relevant images based on a selected group of images. The user can either accept some of the suggested images by dragging them into the current group, or simply ignore the recommendations. There are a few constraints to the recommendation system that arise from the application. First, no image that is already contained in the group should be recommended again. Second, since organisation and interaction with the interface are the primary concern, the recommendations should be limited to a small number of images presented close to the location of the group on the workspace (see Fig. 1). So as not to burden the user, the number of recommended images is based on the standard cognitive limits of $7 \pm 2$ [27].

The system can adapt its recommendations based on learning the features that images in a group have in common and observing user actions and preferences over time. When new images are inserted in the group, the system updates its learning parameters in order to improve its future recommendations. Since the user ultimately decides on group memberships, the groups reflect the current semantics in the context of usage of the image collection. The recommendation system we will describe next is based on learning the similarities of images. Future work will improve recommendation quality by using contextual and usage information to better capture semantic information.

In brief, the advantages of EGO are the following:

- The interactive grouping is a flexible means to communicate both short- and long-term, specific and multifaceted information needs.
- The query formulation problem is reduced significantly by supporting an interaction metaphor for traditional ways of information management.
- The semantic gap is narrowed by the abstraction to high-level semantic groupings, reflecting an individual's task-specific mental model of the data.
- The user leaves trails of their actions behind, that not only the system can exploit for adaptation but also other people can trace. Hence, EGO is ideal in a collaborative work context.

The next sections will deal with the implementation details of EGO: the way images are represented in the system and the matching and recommendation algorithms.

### 4.2 Image Representation

The images are represented according to the hierarchical object model proposed in [10]. This model makes a distinction between the various visual features extracted. Rather than representing an image by a single stacked feature vector, it is composed of a set of feature vectors, one for each distinct feature implemented.

**Distance Measure** The distance between an object $x$ in the database and a given query representation $q$ is computed in two steps. First, the individual feature distances $g_i$ (for $i$ in $1..I$, where $I$ is the number of features) are computed by the generalised Euclidean distance,

$$g_i = (\boldsymbol{q_i} - \boldsymbol{x_i})^T W_i (\boldsymbol{q_i} - \boldsymbol{x_i}) \tag{1}$$

where $\boldsymbol{q_i}$ and $\boldsymbol{x_i}$ are the $i$-th feature vectors of the query $q$ and the database object $x$ respectively, and $W_i$ the *feature transformation matrix* used for weighting the feature components. $W_i$ is a $K_i \times K_i$ real symmetric full matrix, where $K_i$ is the $i$-th feature dimension. The second step is then to combine the individual distances to arrive at a single distance value $d$. This is achieved by a linear combination between $\boldsymbol{g} = [g_1, ..., g_I]^T$ and a feature weight vector $\boldsymbol{u}$,

$$d = \boldsymbol{u}^T \boldsymbol{g} \tag{2}$$

**Implemented Features** We use the following 6 low-level colour, texture and shape features (feature dimension): Average RGB (3), Colour Moments (9) [28]; Co-occurrence (20), Autocorrelation (25), Edge Frequency (25) [29]; Invariant Moments (7) [30].

### 4.3 The Recommendation System

The recommendation system is based on a relevance feedback algorithm, that attempts to learn the best query representation and feature weighting for a selected group of images. As far as the learning system is concerned, each group image is regarded as a positive training sample. The proposed group-based learning scheme involves (1) updating the system's matching parameters, (2) creating a multi-point query representation and computing a ranked list for each query point based on the learnt parameters, and (3) combining the individual result lists for the new recommendations. We will focus on each of these steps in more detail in the following.

**Learning the Feature Weights (1)** The parameter adaptation is achieved by finding new feature weights based on the feedback samples. We adopt the optimised framework for learning the feature weights proposed in [11]. Due to the hierarchical object model, it distinguishes between intra- and inter-feature weights. The optimal intra-feature component weights are given by an optimal feature space transformation matrix $W_i$. $W_i$ is calculated as,

$$W_i = \det(C_i)^{\frac{1}{K_i}} C_i^{-1} \tag{3}$$

where $C_i$ is the *weighted covariance matrix* of the $N$ positive examples according to the $i$-th feature. $W_i$ takes the form of a full matrix, if $N$ is larger than the dimensionality of the feature, otherwise only the diagonal entries are considered. The optimal inter-feature weights $\boldsymbol{u} = [u_1, .., u_I]$ are the weights that best capture the inter-similarity between the training samples. The $\boldsymbol{u_i}$'s are solved by,

$$u_i = \sum_{j=1}^{I} \sqrt{\frac{f_j}{f_i}} \tag{4}$$

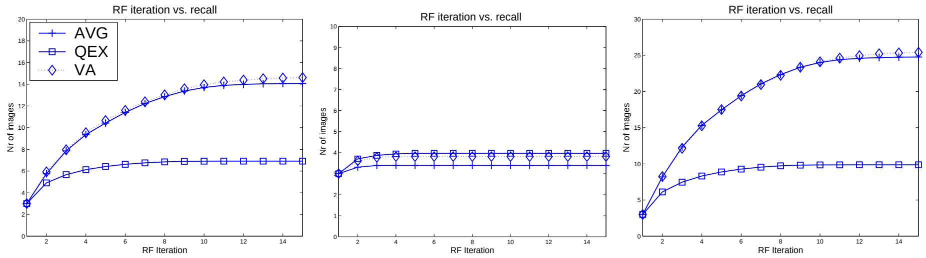
where $f_i = \sum_{n=1}^{N} \pi_n g_{ni}$. The optimal intra-feature weights $W_i$ and the optimal inter-feature weights $u$ are used in Equations (1) and (2) respectively to calculate the total distance between a database object and the query representation.

**Computing the Query Representation (2)** Instead of calculating one overall query representation as in [11], our scheme relies on a form of query expansion. The chosen query representation for a group is a multi-point query [12], whereby each query point represents one cluster of visually similar images in the group. The clusters are computed by an agglomerative hierarchical clustering algorithm, using Ward's minimum variance criterion [31]. The ideal number of clusters is automatically estimated using the method presented in [32]. The query points are selected as the image closest to each cluster centroid. Each query point is associated with a weight relative to the cluster size, ie $w_i = \frac{n_i}{N}$, where $n_i$ is the number of images in cluster $i$, and $N$ the total number of images in the group.

**List Combination (3)** When issuing the multi-point query to the system, a separate result list will be returned for each query point, which need to be combined. An ongoing investigation of several combination strategies has led us to choosing the rank-based *voting approach (VA)*. In this approach each query representative is treated as a voter producing their own individual orderings of candidates (images) based on the similarity to this query. The combined list is computed based on an adaptation of the *median rank aggregation* method as in [33], which sorts the database objects with respect to their median of the ranks they receive from the voters. This algorithm is very efficient and database friendly. The idea can be sketched as follows. Assume each voter produces a ranked list. From each list, access one element at a time, until a candidate is encountered in the majority of the lists, place this candidate as the top ranked of the final list. The second candidate will be placed second top, and so on. Continue until top $k$ candidates are found, or there are no more candidates. To incorporate the query-point weights, $w_i$, determined above, each list, $l_i$ (where $1 \leq i \leq L$ and $L$ the number of voters), is able to score its candidates by its weight. The overall score of a candidate, $s_c$, is accumulated: $s_c = \sum_{i=1}^{l} w_i$, where $l \leq L$. The majority criterion is fulfilled, if $s_c > 0.5$ (this candidate is seen in the weighted majority of lists). Further, the lists are sorted in descending order of their weights, as this algorithm is sensitive to the sequence in which they are processed.

### 4.4 Evaluation

We have chosen the recommendation algorithm just described based on an ongoing performance evaluation of various alternative techniques, which will be described elsewhere. The evaluation is based on Photo CD7 of the Corel collection, containing 238 categories of ca. 100 images each. 10 categories are chosen, which reflect high-level semantic concepts, such as "roses", "flags", "tribal people", "aviation", etc. User interaction is simulated by starting with groups of 3 randomly chosen images from a given category and performing pseudo-relevance feedback from the top 10 returned images until no more relevant images can be found. Preliminary results show that the proposed algorithm (VA) performs significantly better than the multi-point query technique (QEX) of [12], which




(a) all categories (b) heterogeneous categories (c) homogeneous categories

**Fig. 2.** Average number of images found per pseudo-relevance feedback iteration

simply combines the individual scores linearly. VA also slightly outperforms the baseline method (AVG) of [11], where only one query point is determined as the total average of all group images. Figure 2(a) shows the results for the simulated run just described. The graph depicts the average number of images found in each iteration, based on 50 queries per category.

Analysing the individual categories, we could identify two classes, namely *homogeneous* and *heterogeneous* categories. Homogeneous categories contain visually similar images and are well distinguishable from other categories (eg "roses"), while heterogeneous categories contain visually less similar images and are not easily distinguishable from other categories (eg "tribal people"). Our sample categories contained 5 of each. The results for the heterogeneous categories are displayed in 2(b), while 2(c) depicts the homogeneous categories. It shows that AVG performs very well on homogeneous categories, while it performs slightly worse than the multi-point queries on heterogeneous categories. However, VA manages to capture a group's query representation well in both circumstances. It should be noted that the majority of queries (about 1/3) in heterogeneous categories does not manage to find any relevant images in the first iteration, starting with only 3 images. AVG is particularly poor with 39% of these queries, compared to 28% and 29% for QEX and VA, respectively.

Ultimately, the effectiveness of EGO can only be investigated in an evaluation involving real users. Such an evaluation is planned after all the components of the system have been tested individually.

## 5 Future Work

The development of EGO is still in its early stages. We are currently working on various aspects of the system that are needed in order to create a versatile, adaptive, personalised multimedia management tool. The following features are work-in-progress:

- **Enhanced search facilities**:
  As mentioned before, the search component could be enhanced by supporting additional query strategies and alternative result visualisation techniques.

- **Textual annotations**:
  Textual annotation as a means to narrow the semantic gap are desirable. The interface has potential for a group-based annotation operation, which is both simple and efficient, as has been shown in ImageGrouper [13]. Starting from these user-provided annotations, a learning framework as in [7] can be employed to propagate labels.
- **Contextual information**:
  Collecting usage and contextual information is one of our main objectives for EGO. Usage information will include access counts of images and groups, change rates of groups, etc. The contextual information under consideration arises from the fact that images can belong to multiple groups. The context of the groups an image belongs to will be used in the recommendation system.
- **Linkage of groups**:
  So far, the system described has hardly any browsing facilities. To remedy this shortcoming, we envisage a browsable workspace by creating links between groups. The links will be based on similarity of groups (both feature similarity as well as usage similarity) and will be visualised by arrows connecting two groups. The strength of a link (ie similarity score) will be reflected by the arrow's thickness and length. The resulting network of linked groups provides the user the possibility to navigate the "group-space".
- **Extension to other data formats**:
  Finally, an extension to multimedia data could be possible. This could include an integration of text documents based on mechanisms studied in Fetch [24], and video data by adopting more general representation strategies of multimedia retrieval tools.

## 6 Discussion and Conclusions

We have proposed and developed EGO, a tool for personalised multimedia management. In this paper, we have described its interface and the recommendation system that allows it to adapt to its users. The mechanism underlying the recommendation system is based on learning multi-point queries and a voting approach, which we have proposed to combine the evidence from the query points. If we reflect back on the issues mentioned as the primary motivations for the design process, we can see how EGO addresses the questions raised previously:

- *"What is the meaning of an image?"*
  We do not claim we have solved the problem of automatically determining an image's meaning. But as argued before, successful approaches have to recognise the importance of the context, which is not within the retrieval engine, but is determined by the tasks and work environment. To truly make an effort towards cracking the meaning, the image has to be placed in the wider context. In EGO the semantics in the images are conveyed through groupings that the user creates over the course of time. The organisation resulting from the long-term interaction reflect the usage of the collection in

the user's context. From this organisation, it is easier for the system to infer the intended semantic meaning.

– *"How can the user be assisted in communicating their information need?"*
  EGO does not require the user to think in terms of the system (ie how to formulate a query, how a search works, etc) but engages in an iterative organisation process that iteratively defines their semantic needs. This process is closer to every-day solutions of managing information, hence affording traditional problem solving techniques and natural ways of communicating their information need.
– *"How can the time-varying nature of information needs be modelled in the system?"*
  EGO invites the user to create groups according to the multiple facets of their need. The groups are created and changed over multiple sessions, so that they capture aspects of the user's long-term need. Organised on a workspace, they leave behind trails of actions used by the system to adapt to the user's need and enabling users to trace and reflect on their actions.

To conclude, the design of EGO as a tool to create a task-specific organisation of images reflecting an individual's mental model, will overcome many of the problems of traditional CBIR systems.

## References


1. Bates, M.J.: The design of browsing and berrypicking techniques for the online search interface. Online Review **13** (1989) 407–424
2. Ingwersen, P.: Information Retrieval Interaction. Taylor Graham, London (1992)
3. Picard, R.W.: Affective Computing. MIT Press, Cambridge, MA (1997)
4. Belkin, N.J.: Interface techniques for making searching for information more effective. In: CHI'2003, Best Practices and Future Visions for Search UIs. (2003)
5. Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Analysis and Machine Intelligence **22** (2000) 1349–1380
6. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval. (2003)
7. Zhou, X.S., Huang, T.S.: Unifying keywords and visual contents in image retrieval. IEEE Multimedia **9** (2002) 23–33
8. Santini, S., Gupta, A., Jain, R.: Emergent semantics through interaction in image databases. IEEE Trans. Knowledge and Data Engineering **13** (2001) 337–351
9. ter Hofstede, A.H.M., Proper, H.A., van der Weide, T.P.: Query formulation as an information retrieval problem. The Computer Journal **39** (1996) 255–274
10. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance feedback: A power tool for interactive content-based image retrieval. IEEE Trans. Circuits Syst. Video Technol. **8** (1998) 644–655
11. Rui, Y., Huang, T.S.: Optimizing learning in image retrieval. In: IEEE Proc. of Conf. on Computer Vision and Pattern Recognition (2000) 236–245
12. Porkaew, K., Chakrabarti, K., Mehrotra, S.: Query refinement for multimedia similarity retrieval in MARS. In: Proc. of the ACM Multimedia Conf. (1999)

13. Nakazato, M., Manola, L., Huang, T.S.: ImageGrouper: Search, annotate and organize images by groups. In: Proc. of the VISual Conf. (2002)
14. Zhou, X.S., Huang, T.: Relevance feedback in image retrieval: A comprehensive review. ACM Multimedia Systems Journal **8** (2003) 536–544
15. Urban, J., Jose, J.M., van Rijsbergen, C.J.: An adaptive approach towards content-based image retrieval. In: Proc. of the 3rd Int. Workshop on Content-Based Multimedia Indexing (CBMI'03) (2003) 119–126
16. White, R.W., Jose, J.M., Ruthven, I.: An approach for implicitly detecting information needs. In: Proc. of the Int. Conf. on Inf. and Knowledge Management. (2003)
17. Campbell, I., van Rijsbergen, C.J.: The ostensive model of developing information needs. In: Proc. of the Int. Conf. on Conceptions of Library and Informaion Science. (1996) 251–268
18. Garber, S.R., Grunes, M.B.: The art of search: A study of art directors. In: Proc. of the ACM Int. Conf. on Human Factors in Computing Systems. (1992) 157–163
19. Markkula, M., Sormunen, E.: End-user searching challenges indexing practices in the digital newspaper photo archive. Information Retrieval **1** (2000) 259–285
20. Santini, S., Jain, R.: Integrated browsing and querying for image databases. IEEE Trans. Multimedia **7** (2000) 26–39
21. Chen, J.Y., Bouman, C.A., Dalton, J.C.: Hierarchical browsing and search of large image databases. IEEE Trans. Image Processing **9** (2000)
22. Rodden, K., Basalaj, W., Sinclair, D., Wood, K.: Does organisation by similarity assist image browsing? In: Proc. of the ACM Int. Conf. on Human Factors in Computing Systems. (2001) 190–197
23. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. Computer **28** (1995) 23–32
24. Martin, I., Jose, J.M.: Fetch: A personalised information retrieval tool. In: Proc. of the RIAO'2004 Conf. (2004)
25. Kang, H., Shneiderman, B.: Mediafinder: An interface for dynamic personal media management with semantic regions. In: Proc. of the ACM Int. Conf. on Human Factors in Computing Systems. (2003) 764–765
26. Lim, J.H.: Explicit query formulation with visual keywords. In: Proc. of the ACM Multimedia Conf. (2000) 407–409
27. Miller, G.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. The Psychological Review **63** (1956) 81–97
28. Stricker, M., Orengo, M.: Similarity of color images. In: Proc. of the SPIE: Storage and Retrieval for Image and Video Databases. Volume 2420. (1995) 381–392
29. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. 2nd edn. Brooks and Cole Publishing (1998)
30. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory **IT-8** (1962) 179–187
31. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press (1999)
32. Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. Technical Report CS-2003-18, Florida Institute of Technology (2003)
33. Fagin, R., Kumar, R., Sivakumar, D.: Efficient similarity search and classification via rank aggregation. In: Proc. of the ACM Int. Conf. on Management of Data. (2003) 301–312