Multimedia Tools and Applications, 15, 83–103, 2001

Filter Image Browsing: Interactive Image Retrieval by Using Database Overviews

JEROEN VENDRIG vendrig@science.uva.nl
MARCEL WORRING worring@science.uva.nl
ARNOLD W.M. SMEULDERS smeulders@science.uva.nl
Intelligent Sensory Information Systems (ISIS)[*]*, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands*

Abstract. Human-computer interaction is a decisive factor in effective content-based access to large image repositories. In current image retrieval systems the user refines his query by selecting example images from a relevance ranking. Since the top ranked images are all similar, user feedback often results in rearrangement of the presented images only.

For better incorporation of user interaction in the retrieval process, we have developed the Filter Image Browsing method. It also uses feedback through image selection. However, it is based on differences between images rather than similarities. Filter Image Browsing presents overviews of relevant parts of the database to users. Through interaction users then zoom in on parts of the image collection. By repeatedly limiting the information space, the user quickly ends up with a small amount of relevant images. The method can easily be extended for the retrieval of multimedia objects.

For evaluation of the Filter Image Browsing retrieval concept, a user simulation is applied to a pictorial database containing 10,000 images acquired from the World Wide Web by a search robot. The simulation incorporates uncertainty in the definition of the information need by users. Results show Filter Image Browsing outperforms plain interactive similarity ranking in required effort from the user. Also, the method produces predictable results for retrieval sessions, so that the user quickly knows if a successful session is possible at all. Furthermore, the simulations show the overview techniques are suited for applications such as hand-held devices where screen space is limited.

Keywords: image retrieval, human-computer interaction, query by visual example, similarity-based query, performance evaluation

1. Introduction

Large digital image archives require access methods for aiding the user in the search for images. The shift from retrieval of structured data as in conventional databases to unstructured data calls for new techniques and interaction concepts. In most current systems, however, success depends on constraints posed on the applications. Choosing limited domains makes it possible to use domain specific image characteristics, while the use of small data sets allows the employment of sophisticated but computationally expensive features. Since in practice large image databases consist of the union of several small domains, downscaling from large and general domains to small specific sub-domains is desired so that available image features can be fully exploited.

[*]http://www.wins.uva.nl/research/isis/.

User interaction, exploiting human knowledge about the query and context, can be helpful to achieve downscaling. According to the image retrieval overview article of Rui et al. [13] one of the most important future research directions is the introduction of the "human in the loop."

In an interactive system the overview of database content in initial and intermediate stages plays an important role. In traditional image retrieval systems the overview consists of the top of a relevance ranking, which is considered to be the answer to the user's information need. From a conceptual perspective, additional user interaction is interpreted as a new query.

Recent research has taken more interest in query refinement and relevance feedback. There are at least two reasons for this development. One reason is the possibility of extracting knowledge from relations between the selected images to improve relevance ranking [14]. Another reason is the insight human-computer interaction does not only involve user clicks, but also system's feedback about the content of the database so that users are able to specify their information need more precisely.

For many systems relevance feedback just means the user judges the relevance of example images shown on the screen. However, as argued in [2] relevance feedback should not only lead to refinement of the query, but also to refinement of the system's answer. The system is said to refine its answer, or in other words "to be adaptive" [12], when the selection of a presentation set is based on all input of the user, not just the last selection. We call this *vertical* relevance feedback, as opposed to *horizontal* relevance feedback, which is based on the correlation between images selected in one particular state. Examples of horizontal relevance feedback can be found in MARS [14], WebSEEk [17] and El niño [16]. The former two systems adapt the weights of features. The latter system adapts the similarity criterion.

In our opinion, vertical relevance feedback is essential for interactive systems. Therefore, in this paper image retrieval is considered from a session based interaction point of view. In Section 2 existing retrieval methods are classified, described and evaluated in the context of user interaction. In Section 3 the new method Filter Image Browsing, combining and extending the strengths of two existing retrieval methods, is presented. In Section 4 an evaluation method based on user simulations is described. Results are presented in Section 5. Conclusions are given in Section 6.

2. Image retrieval systems

2.1. Classification

For image retrieval systems with large datasets, varying approaches are found in literature. A good overview of query types (depending on the kind of information need for a user) and interface interaction types (the way a query is expressed by a user) is given in [10]. However, both query type and interface interaction type concern man-machine interaction and are related to a large extent. We therefore combine elements of both lists into one classification of image retrieval systems according to the role of user input. It determines the interaction type from the user's point of view. The classification contains three types of image retrieval systems, with seven subclassifications in total.

- *Query by Abstraction*. The user input consists of facts in a predefined fixed format.
  Classifying abstractions according to the way they are derived, as done in [9], results in two subclasses.

  – Query by *Symbolic Descriptions*. The user input consists of abstractions that are information-bearing attributes. They are derived from the context and interpretation of an image.
  – Query by *Image Features*. The user input consists of abstractions that are non-information-bearing feature values, which can be derived directly from the image data.

- *Query by Pictorial Example* (QPE). The user input consists of one or more images. The system returns a relevance ranking, i.e., an ordered list containing the similarity scores of the user input images to the images in the database.
  There are three ways for a user to provide an example image as input to the system.

  – Query by *external* Pictorial Example. The input image is acquired from sources outside the system.
  – Query by *internal* Pictorial Example (QiPE). The input image is selected from the image database.
  – Query by *Construction*, also known as Query by Canvas or Query by Sketch. The input image is synthetic, constructed by the user.

  The choice for a subtype of Query by Pictorial Example depends on the skills of the user and the domain of the information system.
- *Query by Navigation*. The user input consists of a choice of one of the navigation controls provided by the system.
  The input can be classified further based on the relation of the navigation controls to the content of the image database.

  – Query by *Association*. The navigation controls depend on the content of images or metadata in the database. Choice for a predefined control leads the user to related content.
  – *Visual Inspection*. The navigation controls are content-independent with respect to the database. Choice for a predefined control leads the user to another set of images that is not intentionally related with respect to content.

Current image retrieval research focuses primarily on Query by Pictorial Example. Therefore, the method is elucidated in the following paragraphs. The three subtypes of QPE are combined in most systems, because the underlying techniques are similar. For example, the Virage Image Search Engine [1] can be used in applications of all three methods. Still, the type of user input heavily influences the applications in which each type is most effective.

For Query by external Pictorial Example, the user has to have an image at his disposal to feed to the system. This is found in many systems, e.g., in QBIC [6] and PicToSeek [7]. Query by external Pictorial Example systems are usually used to retrieve other information than pictorial data.

Query by internal Pictorial Example does not require any special skills of the user, but does instead depend on the feedback of the system. Systems as QBIC [6] and Photobook [11] employ internal pictorial examples. Query by internal Pictorial Example is mostly used when suitable query images from outside the system are not available.

In Query by Construction a pictorial abstraction of the desired image is synthesized. It may contain parts of real images. This concept is implemented in e.g., QBIC and WebSeek [18]. Query by Construction heavily depends on the artistic skills of the user.

As can be concluded from the appearance of example systems in various classes, in practice image retrieval systems make use of more than one retrieval method. In the next paragraph the relation between the input and output of the various classes of image retrieval systems will be shown, so that user interaction in integrated systems can be analyzed.

2.2. Transactions diagram

Complex image retrieval systems use several methods as building blocks for the entire system. In this section it is determined what building blocks contribute to user interaction in a retrieval session by specifying the input and output types of the retrieval methods described in Section 2.1. A retrieval session is the collection of subsequent information requests concerning one particular information need of the user.

The seven retrieval methods are divided into two types, viz. proactive and reactive, based on the moment the method requires user input. Proactive methods provide the user with information about the database to start the retrieval session, while reactive methods await user input and then process the query.

In the Image Retrieval System Transactions diagram (figure 1), the input and output of the retrieval methods and their relations are shown. The initial input for proactive methods is empty. However, proactive methods can have images as input that are internal to the system, i.e., present in the database. The initial input needed for reactive methods depends on the specific method. All methods have images (from the database) as output. In addition, Visual Inspection and Query by Navigation produce controls as output.

Three methods, viz. Visual Inspection, Query by Navigation and Query by internal Pictorial Example, can have their own output as input. In the diagram this is represented in

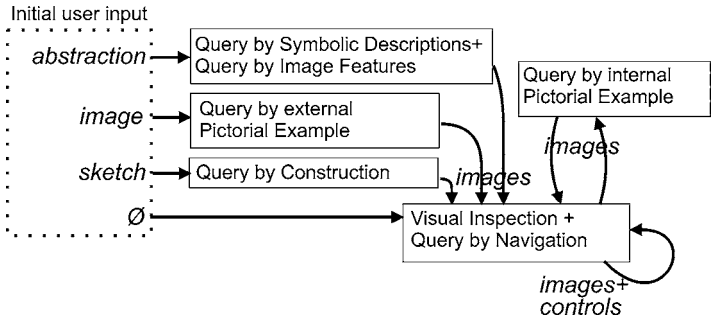

Figure 1. Image Retrieval System Transactions diagram.

the form of a loop. Each cycle in such a loop requires interaction with the user to adapt the query. Visual Inspection is always used, since it is required for every system in which a human user evaluates images. However, for large image collections this method is too time consuming for the user. Query by Association and QiPE are the other two methods that can be used more than once in an interactive retrieval session. To determine their usefulness in interaction, the characteristics of these two methods are explored in more detail.

The advantage of Query by Association is the explicit use of structure. Structure can easily be visualized, e.g., by showing the hierarchical or network relations between images. The disadvantage is the required construction of the relations between the items in the database. Most implementations of Query by Association use manually annotated links to overviews and images. This is unfeasible for large collections.

Query by internal Pictorial Example has the advantage of being very flexible and dynamic. The search path is determined runtime by computing which images are most similar to a given example. There are however two important drawbacks to the QiPE method. Firstly, the construction of the set of images shown to the user at the start of a retrieval session, the initial overview, is non-trivial. It should be a concise summary of the database content. Most QiPE systems, however, select just a set of random images from the database. Secondly, users can get stuck at a local optimum when selecting example images presented by a system, here called premature convergence. Premature convergence prevents a user from incrementally specifying his information need. In the case of QiPE, convergence is premature when the images shown do not yet have enough similarity to the images desired by the user, but do not lead to new rankings anymore.

Both Query by Association and Query by internal Pictorial Example contain powerful concepts for interactive systems. The former is structured and comprehensible, the latter is dynamic and adaptive. As both methods provide user guidance through an image collection, they can be seen as special cases of one another. Their strengths are combined in the Filter Image Browsing method described in the next section.

3. Filter image browsing

In Filter Image Browsing a user recursively selects dynamically generated clusters of images. By choosing the cluster most similar to the information need, the user zooms in on a small collection of relevant images. The scatter/gather method [4] uses a comparable approach for the retrieval of textual documents, but focuses on the visualization of features of document clusters in the form of keywords. The goal of Filter Image Browsing is to allow quick retrieval of images by facilitating interaction between system and user by means of database overviews.

Filter Image Browsing can be seen as a structuring overlay over Query by internal Pictorial Example. The structuring overlay handles the overview problem that exists in traditional QiPE systems. Alternatively, Filter Image Browsing can be viewed as the addition of a dynamic zoom function for image databases to Query by Association. In Section 3.1, the method and its consequences are considered. The ranking techniques used for Filter Image Browsing are discussed in Section 3.2.

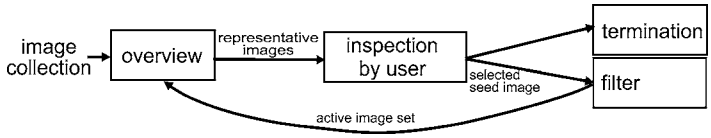

Figure 2. Filter Image Browsing retrieval process.

3.1. The retrieval process

Filter Image Browsing (see also figure 2) first presents an initial, small, overview of the entire content of the database in the form of a set of images to the user. The user inspects the images shown and selects the images most similar to the images he is looking for. Then the system performs a runtime similarity ranking with respect to the selected images. Next, the filter step, which characterizes this method, is performed. Only the images most alike the query image are used in the remainder of the retrieval session. This is a fixed percentage of the currently active image set. The other images are excluded from further similarity rankings.

The three steps overview, selection, and reduction are repeated until the set of remaining images is so small that it is feasible for a user to switch to Visual Inspection. The impact of the reduction step is significant. The set of images shown to the user will not converge prematurely, because it does not consist of just the top most similar images. Instead, each overview is based on the subset of images in that particular state of the retrieval process. Thus the user zooms in on the image database and is able to refine his query iteratively.

The use of reduction filters potentially results in the loss of relevant images during the retrieval process. Once an image is excluded, it cannot be retrieved anymore during that particular session. This problem is avoided by balancing the amount of reduction and the number of required selections. Then the problem is limited to the quality of the similarity ranking operator only.

The user can choose how much of his time he wants to invest in interactions to get better results. In traditional systems users may not want to invest much time in browsing because they get lost in the information space or get stuck in a local optimum. In Filter Image Browsing, however, we expect the user to be willing to invest time because the amount of interactions needed is predictable and because feedback in the form of overviews shows the user's interaction has impact on the system.

Now let us consider the whole process in more detail. Let I be the set of images in the database. I_s is the active set of images in state s of the retrieval process. In the initial state $\mathrm{s} = 0$ and $\mathrm{I}_0 = \mathrm{I}$. I is assumed to be static, i.e., insertions, updates and deletions of entities in the database are not allowed during a retrieval session. A retrieval session lasts from the start-up of the retrieval system until the (user initiated) termination of the system.

In each state, the system constructs a presentation set $\mathcal{I}_s \subset \mathrm{I}_s$ from which the user selects one or more example images, which are used to reduce I_s to I_{s+1}. On the new active set I_{s+1} once again the overview operation is applied to provide the user with a new overview. The cycle of overview, image choice and filtering is repeated until the number of elements in I_{s+1} reaches a predefined limit N_{end}, or until the user decides to end the retrieval session. He

will normally terminate the system when the desired images are found, or when he has the impression the images cannot be found. The latter case indicates that either the interactive system failed or the desired images are not present in the database.

In the following paragraphs the overview and filter operations are explored in more detail. As we will show later, the overview operation is influenced by the filter operation. Therefore, the filter operation is described first.

3.1.1. Filter. In this context, the appliance of a filter results in a change of scope of the active image set. Since the purpose of Filter Image Browsing is to zoom in on a set of images that suits the user's information need, reduction is the most important filter operation. It would also be possible to use expansion for a filter operation after the reduction filter has been applied at least once. However, to limit the number of navigational choices, we here focus on reductions only.

The goal of a reduction is to achieve a smaller set of images still containing the images desired by the user. Since the system does not know which images are desired, it has to rely on the choices the user made to fulfill the goal. Similarity ranking of the active set according to the user selected images is an appropriate technique to select images for the new (reduced) set. I_{s+1} then contains the images from I_s most similar to the query images i. For this purpose, a reduction factor ρ is used. For example, if $\rho = 0.25$, I_{s+1} will be four times as small as I_s. The reduction factor can be adjusted in each state, based upon the confidence of the user in his choice or the speed with which the user desires to browse through the collection.

The filter process is given by:

$$I_{s+1} = \varphi_{\Delta}(I_s, i, \rho) \text{ where } I_0 = I \tag{1}$$

φ is a k-nearest neighbor function where $k = \lceil \rho \cdot N_s \rceil$. φ uses the scoring function Δ described in Section 3.2. The use of a dynamic scoring function rather than a static cluster method allows runtime adaptation of all parameters in a state.

The reduction operation targets at a fixed size for the new active set in order to make a predictable and easy to use system for the user. When all parameters are known, it is easy to compute the number of steps necessary to reduce I_0 to an image set of size at most N_{end}. Since $I_s = \rho^s \cdot |I_0|$, s_{end} for a constant ρ is given by:

$$s_{end} = \frac{\log \frac{N_{end}}{|I_0|}}{\log \rho} \tag{2}$$

3.1.2. Overview. The goal of the overview function is to present to the user a set of images representative for the active image set. The number of presented images must be small, since the user is supposed to look at the images by way of Visual Inspection.

The images in $\mathcal{I}_s$ are the only access to the other images in I_s. In combination with the reduction operation, this results in the problem of orphan images in I_s, i.e., images i for which it is not possible to choose an image from $\mathcal{I}_s$ that results in an I_{s+1} containing i. Since orphan images would always be lost in the next state, their existence has to be prevented by fully covering I_s.

To guarantee I_s is fully covered by $\mathcal{I}_s$, we introduce the *Cover* operation, which enforces the cover constraint. The cover constraint says every image in I_s must be part of at least one of the possible new active sets I_{s+1}.

Cover$(I, \mathcal{I}, N_{show}, \rho)$ extends a small set of seed images $\mathcal{I}$ so that the resulting set covers the active set I. N_{show} is the amount of images suited for on screen presentation. It is set by the system's manager or by the users themselves. *Cover* is constructed such that the resulting set of images X complies with the following conditions:

- $\bigcup_{x \in X} \varphi_\Delta(x, I, \rho) = I$, the cover constraint.
- $|X| \leq N_{show}$, the result has to be small enough to be shown on screen.
- $\mathcal{I} \subseteq X$, the given (small) set of seed images has to be part of the result.
- $X \subset I$, only images from the given active set are allowed.

If no set of images X complying with the given conditions can be found, the system (automatically) or the user (manually) has to adapt either ρ (less reduction) or N_{show} (more images on screen).

Although $\mathcal{I}$ can be an empty set in theory, there are two practical reasons to provide a small set of images as a seed. Firstly, when a seed is given, the number of possible outcomes for X is limited and hence the computational complexity is reduced. Secondly, it is advisable to use the example images, which were selected by the user from $\mathcal{I}_{s-1}$, in the new presentation set, because it provides the user with an anchor point for the evaluation of $\mathcal{I}_s$. The input image set is determined by the *preSelect* operation.

preSelect$(\mathcal{I}, I, N_{show})$ results in a small set of seed images. For example, when $\mathcal{I}$ is empty, *preSelect* can pick a random set of images. In another common example, $\mathcal{I}$ is a set of images selected by the user. *preSelect* is constructed such that result set X complies with the following conditions:

- $|X| \ll N_{show}$, the result set will not dominate the sets it's used for later.
- $\mathcal{I} \subseteq X$
- $X \subset I$

The *postSelect* operation extends a given set of images to a set with a predefined size N_{show}. Although optional in the selection process for a presentation set, the operation is important to ensure a constant amount of output to the user, thereby increasing the predictability of the system. *postSelect*$(\mathcal{I}, I, N_{show})$ is constructed such that result set X complies with the following conditions:

- $|X| = N_{show}$, result set has to be of given size.
- $\mathcal{I} \subseteq X$
- $X \subset I$

Since I_s, N_{show} and ρ are constant in state s, the complete selection process for a presentation set is written in short as:

$$\mathcal{I}_s = postSelect(Cover(preSelect(\mathcal{I}))) \tag{3}$$

In the case of using example images as input for the selection process, $\mathcal{I} \subset \mathcal{I}_{s-1}$.

Implementation of *preSelect* and *postSelect* is trivial. For *Cover* we have constructed a brute force algorithm. Our approach is comparable to the Maximal Marginal Relevance metric (MMR) [8] from the field of text retrieval, which assures "relevant novelty" for a reshuffled ranking. Our approach, however, concentrates on covering all active database objects instead of just the top ranked ones. In our approach, *preSelect* results in a set of one seed image that is either randomly chosen from I_0, or is an image selected by the user from $\mathcal{I}_{s-1}$. The intermediary set C contains all images covered by $\mathcal{I}_s$ in the current stage of the algorithm, $\mathrm{C} \subseteq \mathrm{I}_s$.

1. $\mathcal{I}_s = \emptyset$, $\mathrm{C} = \emptyset$
2. Take seed image i from *preSelect*
3. $\mathcal{I}_s = \mathcal{I}_s \cup \mathrm{i}$
4. $\mathrm{C} = \mathrm{C} \cup \varphi(\mathrm{I}_s, \mathrm{i})$
5. If $\mathrm{C} = \mathrm{I}_s$, cover constraint is fulfilled: exit selection procedure.
6. Choose new seed image i, $\mathrm{i} \in \mathrm{C} \setminus \mathrm{I}_s$. i is chosen such that $\Delta(i, \mathcal{I}_s)$ is maximal
7. Go to step 3.

Constantly switching to the least similar, uncovered seed image in step 6 prevents overlap of the results of the reduction operation. Hence, $\mathcal{I}_s$ emphasizes the differences between images in I_s.

In a best case scenario, $1/\rho$ seed images will be necessary. In practice the brute force approach resulted in an average of 13 images (ranging from minimal 9 to maximal 16 images) necessary to cover our data sets for $\rho = 0.25$. This amount is suitable for desktop applications.

We have implemented two scenarios for the overview function. In one scenario *postSelect* consists of the random selection of images from I_s. In the other scenario the overview techniques described are used, but the cover constraint is not enforced to allow a small sized $\mathcal{I}_s$. The results for both scenarios will be discussed in Section 5.

3.2. Similarity

In principle any ranking method based on a metric distance function can be used in an implementation of the Filter Image Browsing concept. In this section the distribution based ranking function we developed for Filter Image Browsing is described.

To compute an image's position in the ranking, similarity scores (compared to the given image) δ_f of the individual features have to be combined into one value:

$$\Delta = \Gamma_{f \in \mathrm{F}}\, \mathrm{w}_f \cdot \delta_f \tag{4}$$

Γ is the combination operator (sum in our case), F is the total set of features and w is a weighing factor. We use equal weights for each feature for the sake of simplicity, although the FillB concept allows to use different feature sets in each state of the browsing process. The selection of features could be automated, based on computational efficiency, e.g., using costly features only for a small active set, or the feature values of the images in the active set, e.g., if all images are red there is no use for a color feature anymore.

Since the individual features cannot be expected to have similar distributions of their values, the scores themselves cannot be averaged to a meaningful overall score. In [5], Fagin and Wimmers describe a similar problem for the combination of fuzzy logic scoring methods when weights are used. They give a formula for mapping an unweighted collection of scoring functions to a weighted collection. Instead of asking the user for weighting parameters, we add a normalization function to each feature scoring function. Whereas most methods employ normalization of feature values, we normalize based on similarity values. Our normalization process consists of a preprocessing and a query-time step.

In the preprocessing step, for each feature a priori all $N_0 \cdot (N_0 - 1)$ similarity scores between the N_0 images are computed by δ_f, resulting in a similarity score distribution. The distribution is then, again for each feature, sampled into a frequency histogram for similarity scores. If computation of all possible similarity scores is too costly, a random sample of images can be used to speed up preprocessing.

The query-time step is invoked during a retrieval session, each time the original feature scoring function is called. After computation of the similarity score for two images according to the particular feature, the score is looked up in the frequency histogram. A percentile is then returned, e.g., stating the similarity between the objects is in the top 5%. The percentiles of the various features can be compared to one another because they are independent of the distribution and similarity measure used.

The use of the normalization via frequencies does not only allow the use of different types of image features and similarity measures, such as histograms and metric values. It also allows the use of features of other media than images, such as textual context.

Extension of the similarity function to compare a set of given images to an image is straightforward by combining the similarity values for each individual comparison, possibly weighted.

4. Evaluation

In order to evaluate the Filter Image Browsing concept an experiment using the ImageRETRO system[1] (figure 3) was set up. In the following sections we describe the evaluation criteria and the experimental environment.

4.1. Criteria

For evaluation of the effectiveness of Filter Image Browsing, three criteria are used. One for each of the functions Filter Image Browsing is based on, viz. reduction and overview, and one for the overall performance of Filter Image Browsing compared to Query by internal Pictorial Example.

To allow comparison of performance in the various states of the retrieval session, it is assumed the information need of the user is static during the entire session. Thus the assistance of the retrieval system in the definition of an information need cannot be measured. The information need is expressed as a static set of target images T. Furthermore, it is assumed that it is possible to fulfill the information need, meaning $T \subset I_0$.

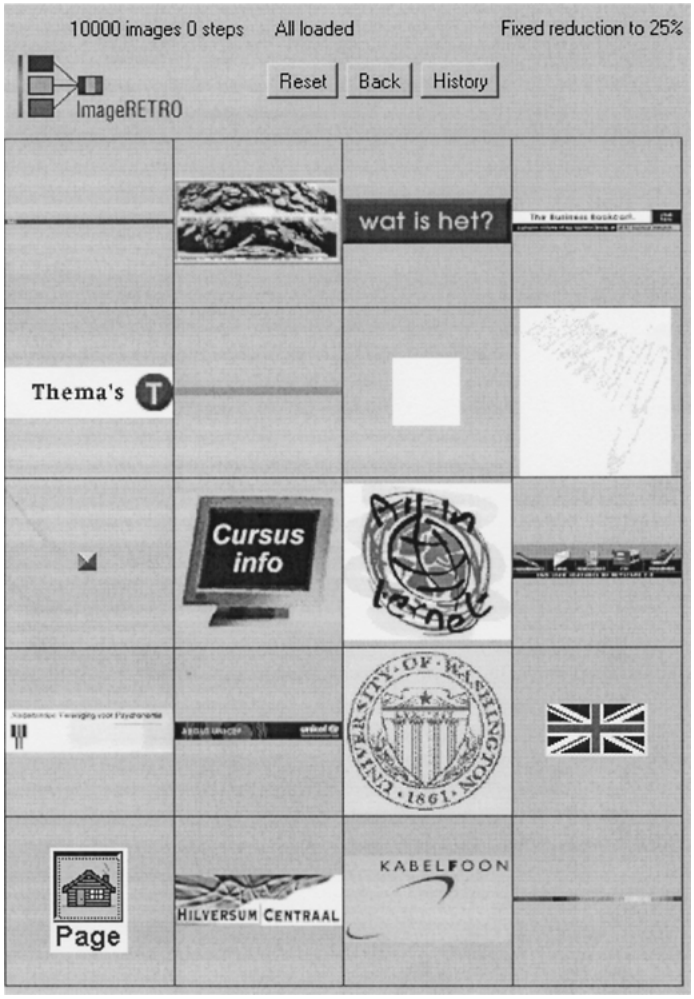

Figure 3. Filter Image Browsing as implemented in the ImageRETRO system. The screendump shows 20 example images forming $\mathcal{I}_0$.

4.1.1. Reduction evaluation. To measure the effect of discarding images during the filter process, the Recall in each state of the retrieval session is computed. The Recall in state s is the number of relevant images in I_s relative to the total number of relevant images in the target set.

$$\text{Recall}_s = \frac{|I_s \bigcap T|}{|T|} \tag{5}$$

Note that the value of Recall is independent of the number of relevant images actually shown to the user via $\mathcal{I}_s$.

In the best case, only irrelevant images are discarded by the reduction operation and Recall is 1 (maximum) in every state. The reduction criterion measures to what degree this best case scenario is approached in practice. Since the loss of relevant images depends on the reduction factor and the resulting size N_s of the active set in state s, Recall is expressed as a function of N_s. By definition, Recall is 1 in the initial state. To prevent forced loss of relevant images, in every state of the retrieval session N_s has to be equal to or greater than the size of T.

In information retrieval studies, the Recall evaluation function is usually accompanied by the Precision function, which measures how many of the retrieved images are relevant. For QiPE, however, the Precision will be constant throughout the entire session and is thus not useful. For FilIB, use of the Precision measure doesn't add much value either, because the only goal of the method is to end up with the entire target set T. Reaching this goal is already measured by Recall in the final state. Intermediate Precision results are not relevant for evaluating FilIB, because the method is based on overviews of the entire active set. If no relevant images are lost, Precision would increase by definition after each reduction. Hence, for FilIB it is sufficient to measure just the loss of relevant images with Recall.

4.1.2. Overview evaluation. For comparison of presentation set generators, Sought Recall SR [15] in each state of the retrieval session is measured. Sought Recall is a variant on Recall as perceived by the users, because it expresses the amount of relevant images the user has actually seen on screen during the current and previous states.

$$\mathrm{SR}_s = \frac{\left| \bigcup_{j=0..s} \left(\mathcal{I}_j \bigcap \mathrm{T} \right) \right|}{|\mathrm{T}|} \tag{6}$$

Filter Image Browsing's cluster overview function assists the user in choosing an example image by presenting a small amount of images representative for distinct parts of the database. This should lead to better results than presentation of randomly picked images. In order to test this hypothesis, SR of the two overview types is compared. For cluster overview to be successful, after convergence to a small I_s its SR has to be higher than SR of the random overview.

4.1.3. Overall evaluation. Since Filter Image Browsing is considered to be an alternative to Query by internal Pictorial Example, the two systems are compared. The SR measure is used in this case as well. Filter Image Browsing has to outperform QiPE within a reasonable amount of user interactions (expressed by the state parameter). For this purpose, SR is expressed as a function of the number of user interactions.

The values for SR can be predicted by using the Recall values, assuming all images in I_s (including the members of T) have an equal chance of being selected for $\mathcal{I}_s$.

$$\mathrm{SR}_s \approx \frac{|\mathcal{I}_s| \cdot \mathrm{Recall}_s}{\mathrm{N}_s} \tag{7}$$

Since SR is known to the user, the prediction could be used to derive Recall. Subsequently, conclusions about continuing or restarting the session can be made.

4.2. Experiment

4.2.1. Domain. A large and diverse domain was chosen to populate the image database. A robot gathered 10,000 images from the World Wide Web by following hyperlinks. The collection of images is representative for the World Wide Web and is described in [19]. It contains digital photographs, icons and other synthetic images, plain alphanumeric labels and combinations of the types mentioned. About 20% of the images can be considered gray valued, the others are colored. The images are available to the public via the World Wide Web.[2]

For every image, index values were computed for 9 simple image features. The features have been chosen to discriminate between groups of images in the domain. The features are based on the distribution of hue, saturation and intensity (HSI), mostly using a combined histogram of hue values (12 bins) and gray values (3 bins). All features are global, i.e., they characterize the entire image rather than a part of it. The concept of region is used, which is defined as a collection of 8-connected pixels with the same hue or gray value. Details about the features can be found in [19].

The features are classified into 3 categories:

- HSI distribution: number of colors, gray/color ratio, and share of dominant color.
- HSI values: average color, color variation, average grayvalue, and average saturation.
- Spatial distribution: number of HSI regions, and number of holes (region surrounded by another region).

4.2.2. Simulation. In the evaluation experiment users were simulated by a user model as introduced in [3]. In the user model it is assumed all users are the same and the decisions they make are based on image features. Hence, the modeled users are consistent and unbiased. Use of a simulation instead of human users prevents the experiment from evaluating human serendipity and experience while evaluation of retrieval method effectiveness is desired. Furthermore, experiments can easily be repeated under exactly the same circumstances by simply resetting parameters.

We note that in our experiment the selection of seed images from the presentation set was based on the same feature values as the system itself uses for computations. Hence it cannot be tested if the similarity computation by the system is the same as the perception of the user. The simulation should therefore only be used to evaluate how systems compare relatively from a user point of view, but not to predict the actual success rate for human users.

Target sets that define the information need of the simulated user have to comply with three conditions:

- Small distance in feature space, so that it is possible in the experiment to cluster the images according to their feature values. The images have to be at most 100-nearest-neighbors of one another. Hence, the choice for the target set depends less on the quality of the features.

- Same style, so that the images are visually similar. The style is defined by objective metadata (a common original site) and a subjective evaluation of the visual content by a human expert.
- Size bandwidth. There is a minimum and maximum for the number of images. Each set has to contain at least 5 images, so that evaluation focuses on finding groups of images instead of one particular image. A group is necessary to define the concept of similar images. The maximum is defined by the number of images shown in the last (converged) state, so that it is possible to reach perfect Recall in each state of the retrieval process.

In practice, there are few image groups complying with all three conditions. Therefore this approach scales to finding ground truths in larger domains if the conditions that can be evaluated automatically are applied first. The subjective evaluation of style is the only condition that requires human efforts and should therefore be applied only after first fulfilling the other conditions.

For the experiment, nine target sets[3] were selected from the collection of 10,000 images. Extracts from four target sets are shown in figure 4.

The algorithm used for simulation of Filter Image Browsing and Query by internal Pictorial Example, make use of the predefined target sets and a given size for $\mathcal{I}_s$(the desired number of pictures to be shown to a user per interaction moment). The latter parameter was fixed on 20 and 5 for the two scenarios in our experiments. Furthermore, Filter Image Browsing uses reduction factor ρ, given as a constant for each retrieval session. Both simulations use the ChooseSeed function, which computes which of the images in the presentation set is most similar to the entire target set. If one of the images in the presentation set is a member of the target set, it is chosen by default. The algorithm is described in pseudo-code.

```
1.  WHILE reasonable amount of user interactions AND NOT converged DO
2.      I_s = getPresentationSet(I_s,i)
3.      Visual Inspection of I_s
4.      IF convergence criterion reached THEN
5.          converged
6.      ELSE
7.          image i = ChooseSeed(I_s)
8.          I_s+1 = getSimilarityRanking(I_s,i)
9.          s = s+1
10. END
```

For QiPE overview $\mathcal{I}_0$ is given. The function getPresentationSet returns overview(I_s,i) for Filter Image Browsing, and the top ranked images from I_s for QiPE. The convergence criterion is "N_s equals presentation size" for Filter Image Browsing, and "high similarity of $\mathcal{I}_s$ and $\mathcal{I}_{s+1}$ (>80% overlap)" for QiPE. Finally, the function getSimilarityRanking results in $\varphi(I_s, i)$ for Filter Image Browsing, and a plain similarity ranking relative to seed image i for QiPE.

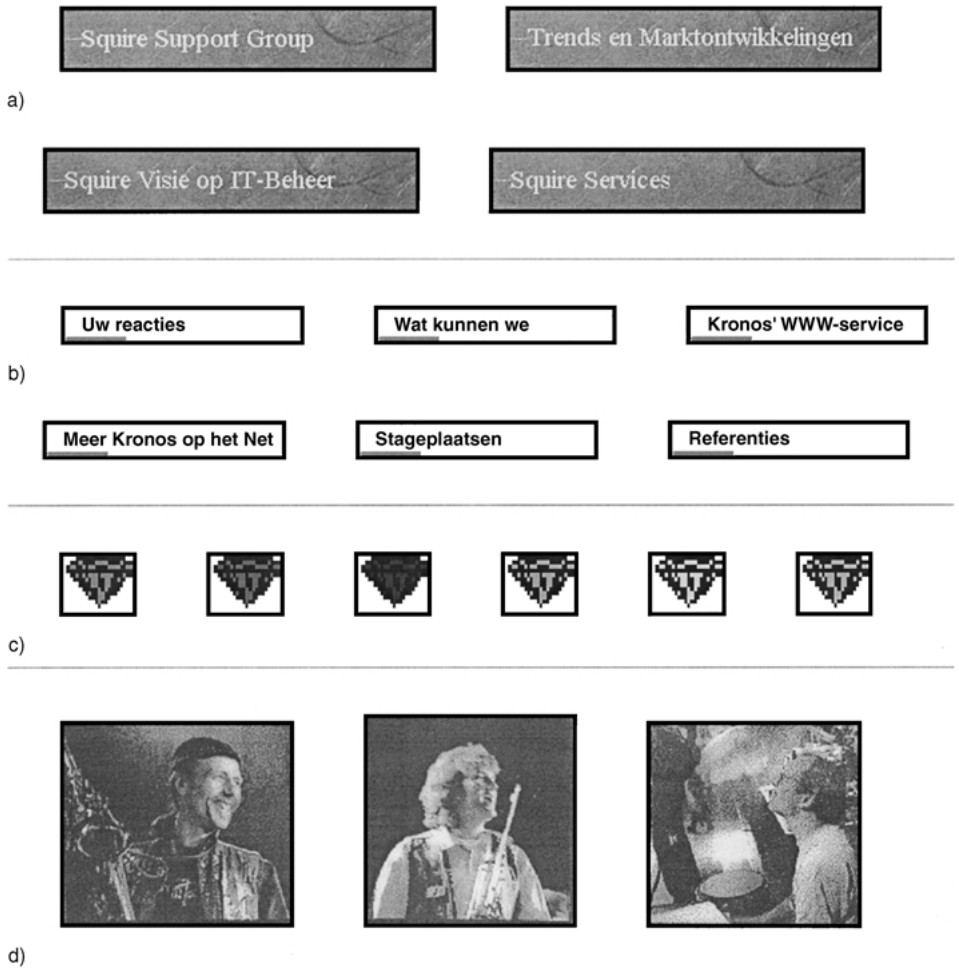

Figure 4. Extracts from four target sets used in the simulation.

5. Results

Simulations were run for the 9 target sets, with 5 different randomized seeds. In Filter Image Browsing simulations, for each of the reduction factors used, the average of 3 slightly varying reduction factors was used to avoid the impact of 'lucky guesses' by the overview function. The reduction factors mentioned in the results are the medians. The results for each Filter Image Browsing simulation are based on 135 runs in total.

In figure 5 the graphs for the evaluation of the reduction operation are shown. The standard deviations for the values are approximately 0.05, with extremes of approximately 0.08 in the final states (size = 20). Note that this figure is not intended to compare the performance of various reduction factors. For each reduction factor a different number of user interactions

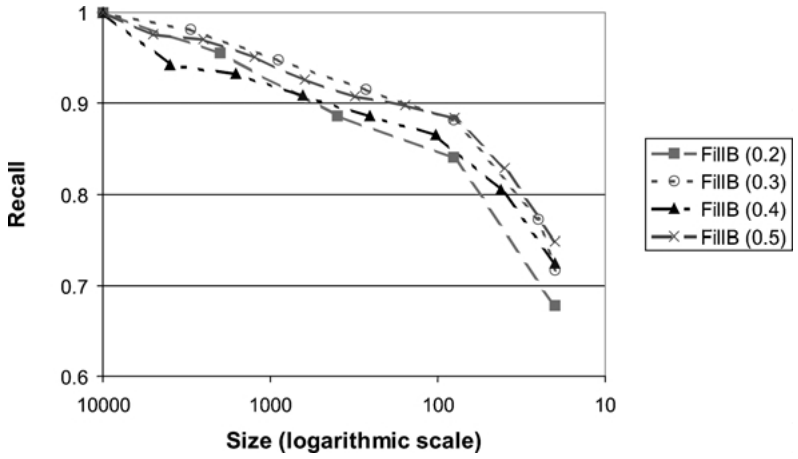

Figure 5. Effect of reduction for various reduction factors.

is required, as the number of markers in a line visualizes. The penalty for the accuracy of higher reduction factors is visible in figure 5. For example, $\rho = 0.5$ results in slightly better performance than $\rho = 0.3$, but it requires three more interactions to converge. The user has to choose between speed (few interactions) and retrieval accuracy.

In Table 1 the results for the evaluation of the overview function are shown. The scenario Cluster (only *postSelect* picks random images) was described in Section 3.1.2. For scenario Random all images in the overview are selected randomly from I_s. For various reduction factors, average SR after convergence (an active set small enough for visual inspection) was measured. The standard deviation is given for comparison of the predictability of the results. The numbers show scenario Cluster outperforms scenario Random for all reduction factors, both in average SR as in standard deviation.

The results from Table 1 vary for each target set. Figure 4 contains samples of both the most and least successful target sets. E.g., in the case of $\rho = 0.3$ the Cluster scenario scored excellent in finding the alphanumeric label images (4a and 4b). The Cluster scenario does have trouble with finding the icons (4c) and photographs (4d) since each picture contains different colors while most features used describe the color content of an image. For the latter target set, results for both scenarios were nearly the same.

The scenario in which only a small amount of images can be shown to the user is evaluated in Table 2. During each overview 5 images are shown to the user. The final, converged image

Table 1. Sought Recall after convergence and its standard deviation over the target sets.

Overview type	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.4$	$\rho = 0.5$
Cluster	0.69 ± 0.07	0.79 ± 0.05	0.77 ± 0.07	0.81 ± 0.08
Random	0.46 ± 0.10	0.51 ± 0.09	0.60 ± 0.10	0.66 ± 0.07

Table 2. Sought Recall for small overview after convergence and its standard deviation over the target sets.

Overview type	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.4$	$\rho = 0.5$
Cluster	0.58 ± 0.07	0.56 ± 0.06	0.55 ± 0.08	0.54 ± 0.09
Random	0.11 ± 0.08	0.12 ± 0.06	0.17 ± 0.08	0.23 ± 0.11

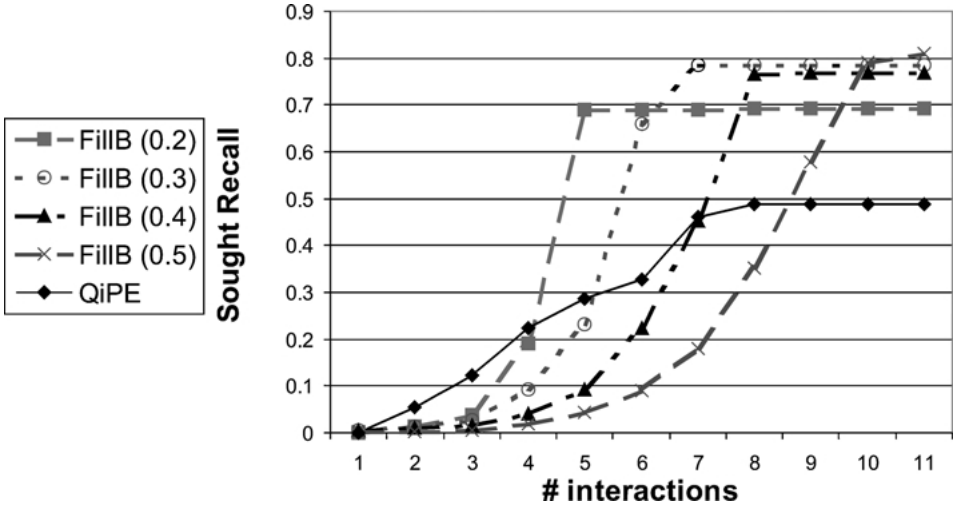

Figure 6. Retrieval performance of Filter Image Browsing (FilIB) and Query by internal Pictorial Example (QiPE).

set shown to the user contains 12 images (the size of the largest target set, so that a SR of 1 can be reached in each simulation). The difference between the results of the Cluster and Random scenarios is significant.

Finally, the comparison of Filter Image Browsing to a Query by internal Pictorial Example system is shown in figure 6. For both systems a cluster based overview was evaluated. In the case of QiPE this means only the initial presentation set is constructed by way of clustering techniques. For Filter Image Browsing four different reduction factors were used. The graphs for Filter Image Browsing simulations are stable when converged. The number of states necessary to converge can be computed from the formula in Section 3.1.1. The maximum number of interactions for both Filter Image Browsing and QiPE is 11.

Standard deviations for FilIB are very low until state 3, but vary from 0.05 up to 0.20 in states 4 to 6. However, after convergence standard deviations are the same as in Table 1. The standard deviation for QiPE measurements increases from approximately 0.06 in state 3 to approximately 0.16 in state 7 and later.

5.1. Discussion

The reduction evaluation shows that even though Filter Image Browsing does cause the loss of desired images, far more irrelevant than relevant images are discarded. A reduction factor of approximately 0.3 can be recommended for Filter Image Browsing, since the retrieval process takes only seven interactions while the recall approximates the recall of higher reduction rates.

The overview evaluation results in Table 1 show that cluster based overview, as implemented in scenario Cluster, is an improvement over the conventional overview with randomly picked images. When the cover constraint is lifted so that it is possible to decrease the size of the overview, the impact of the overview function is even more obvious (Table 2). The scenario with random overview results in poor results, while the scenario with cluster overview still shows reasonable performance. Hence, the overview function is useful for applications allowing the presentation of only a small amount of example images.

The decreasing difference between random and cluster overview results when the size of the overview set increases, leads us to the conclusion that random overview approximates compliance with the cover constraint if it is sufficiently large. However, the cover constraint does not only still result in better Sought Recall, but according to the underlying theory takes outliers into account as well. Hence, the cover constraint provides higher predictability of the retrieval performance because it does not heavily depend on the position of target sets in the feature space.

The overall evaluation in figure 6 shows the user is able to find more relevant images with fewer interactions in a FilIB system than in a QiPE system. However, a FilIB system does require some startup time in the form of user interaction, while a QiPE system provides first results quickly. The curves in the graphs imply that after the first few states, in a QiPE system a local optimum is reached. FilIB on the other hand intends to provide the user with a small converged set of relevant images, which explains the steep graphs and the sudden decrease of the standard deviation in the final state. Since FilIB does not intentionally show relevant images during the overviews, the Sought Recall can vary widely during these states. But in the final, converged state all images in the small set are shown as an end result so that variation in Sought Recall is small.

6. Conclusions

Incorporating user interaction in an image retrieval method as done in Filter Image Browsing pays off. The user feedback consists simply of the selection of an image, while the system's vertical relevance feedback is communicated to the user via the presentation set. Subsequent cycles of database overview and reduction lead the user in few steps to a small collection of relevant images. With regard to the user input classification and the Image Retrieval Systems Transaction diagram, Filter Image Browsing is best comparable to the Query by internal Pictorial Example method.

The simulations used to evaluate the performance of Filter Image Browsing show satisfying results for all determined criteria. We conclude that more elaborate use of user interaction does result in quicker retrieval of images. Further, the results of Filter Image Browsing

are more predictable, as the number of user interactions can be computed a priori. Hence the method is helpful also when the desired images are not present in the image collection, since a user does not have to search indefinitely.

The combination of Query by internal Pictorial Example from the image retrieval field and Query by Association from hypertext research in Filter Image Browsing results in a powerful method for browsing through image databases. Furthermore, the method is useful for mobile computing applications in which only a small amount of images can be shown to the user.

As the ranking method introduced in this paper is independent of data structures and media types, Filter Image Browsing can easily be extended to provide access to the content of multimedia databases.

Acknowledgments

This research was sponsored by MediaLab B.V., Schellinkhout, The Netherlands.

Notes

1. http://carol.wins.uva.nl/~vendrig/imageretro/.
2. http://carol.wins.uva.nl/~vendrig/imageretro/pictures/.
3. http://carol.wins.uva.nl/~vendrig/imageretro/target/targetsets.html.

References

1. J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C.-F. Shu, "Virage image search engine: An open framework for image management," in Proceedings SPIE: Storage and Retrieval for Image and Video Databases IV, San Diego, CA, 1996, pp. 76–87.
2. I.J. Cox, M.L. Miller, S.M. Omohundro, and P.N. Yianilos, "PicHunter: Bayesian relevance feedback for image retrieval," in Proceedings of ICPR '96, Vienna, Austria, 1996, pp. 361–369.
3. I.J. Cox, M.L. Miller, S.M. Omohundro, and P.N. Yianilos, "Target testing and the PicHunter bayesian multimedia retrieval system," in Proceedings of the Advanced Digital Libraries (ADL'96) Forum, Washington D.C., 1996, pp. 66–75.
4. D.R. Cutting, D.R. Karger, J.O. Pedersen, and J.W. Tukey, "Scatter/gather: A cluster-based approach to browsing large document collections," in Proceedings of SIGIR'92, Copenhagen, Denmark, 1992.
5. R. Fagin and E.L. Wimmers, "Incorporating user preferences in multimedia queries," in Proceedings 6th International Conference Database Theory—ICDT '97, F.N. Afrati and P. Kolaitis (Eds.), Delphi: Greece, 1997, pp. 247–261.
6. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," IEEE Computer, Vol. 28, No. 9, pp. 23–32, 1995.
7. T. Gevers and A.W.M. Smeulders, "Pictoseek: A content-based image search engine for the world wide web," in Proceedings of VISUAL'97, San Diego, CA, 1997.
8. J. Goldstein and J. Carbonell, "The use of MMR, diversity-based reranking in document reranking and summarization," in TWLT 14, Language Technology in Multimedia Information Retrieval, D. Hiemstra, F.M.G. De Jong, and K. Netter (Eds.), Enschede: The Netherlands, 1998, pp. 153–166.
9. W.I. Grosky, "Multimedia information systems," IEEE Multimedia, Vol. 1, No. 1, pp. 12–24, 1994.
10. R. Jain, "Infoscopes: Multimedia information systems," in Multimedia Systems and Techniques, B. Furht (Ed.), Kluwer Academic Publishers: Boston, 1996, pp. 217–253.

11. A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," in Multimedia Tools and Applications, B. Furht (Ed.), Kluwer Academic Publishers: Boston, 1996, pp. 43–80.
12. Y. Rubner, C. Tomasi, and L. Guibas, "Adaptive color-image embeddings for database navigation," in Proceedings of the 3rd Asian Conference on Computer Vision (ACCV98), Hong Kong, 1998, pp. 104–111.
13. Y. Rui, T.S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions and open issues," Journal of Visual Communication and Image Representation, Vol. 10, pp. 1–23, 1999.
14. Y. Rui, T.S. Huang, Michael Ortega, and Sharad Mehrotra, "Relevance feedback: A power tool in interactive content-based image retrieval," IEEE Trans on Circuits and Systems for Video Technology, Vol. 8, No. 5, pp. 644–655, 1998.
15. G. Salton and M.J. McGill, "Introduction to Modern Information Retrieval, McGraw-Hill: New York, 1983.
16. S. Santini and R. Jain, "Beyond query by example," in Proceedings of the Sixth ACM International Multimedia Conference, Bristol, England, 1998, pp. 345–350.
17. J. R. Smith and S.-F. Chang, "An image and video search engine for the world-wide web," in Proceedings SPIE: Storage and Retrieval for Image and Video Databases V, San Jose, CA, 1997, pp. 84–95.
18. J.R. Smith and S.-F. Chang, "Visually searching the web for content," IEEE Multimedia, Vol. 4, No. 3, pp. 12–20, 1997.
19. J. Vendrig, M. Worring, and A.W.M. Smeulders, "Filter image browsing," Technical Report 5, Intelligent Sensory Information Systems, Faculty WINS, Universiteit van Amsterdam, http:carol.wins.uva.nl/~vendrig/papers/isis_5.ps.gz, 1998.

Jeroen Vendrig received his M.Sc. degree in Business Information Systems from the University of Amsterdam in 1997. He is currently working towards the Ph.D. degree in the ISIS group. He worked on the interactive image retrieval project with industry partner Origin/MediaLab until 1998. Currently he's working on the "Interactive Exploration of Multi Media Content" project of the Netherlands Organization for Scientific Research (NWO). His research interests include video analysis, the use of context for video database retrieval, and user interaction for building indices and queries on video collections.

Marcel Worring (M.Sc. honors 1988, PhD. 1993) is an assistant professor of computer science at the University of Amsterdam. In the fall of 98, he was a visiting scholar at the Visual Computing Lab, University of California, San Diego.

Main topic of current research is the automatic structuring of the content of multimedia documents to allow for content based access, exploration, and presentation. In this context he is leading a large project in which experimentation platforms for large scale Multimedia Information Analysis (MIA) are being developed. This project is conducted in close relation with industry.

Arnold W.M. Smeulders is professor of Computer Science on Multi Media Information Processing. His interest is in image databases and intelligent interactive image analysis, as well as system engineering aspects of image processing. He is director of the Computer Science Institute and of the Intelligent System Lab of the University of Amsterdam. The ISIS-group conducts research on the theory of computer vision, industrial vision and multimedia information.