The Relevance Pyramid
Combining Browsing and Relevance Feedback in Image Databases

Neeltje Blommestein Theo Gevers
University of Amsterdam
ISIS group
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
gevers@wins.uva.nl
http://carol.wins.uva.nl/∼gevers/

Abstract

In this paper, a new method for image retrieval is proposed. The method combines three existing methods: query-by-example, browsing, and relevance feedback. The method assists users in finding images of the same scene or genre in image databases. First, the user can express query by presenting an example image. Then, the user is given the opportunity to browse through the images, and to refine the initial query in additional relevance feedback loops. The browsing environment consists of a similarity pyramid, in which the different layers represent the database with varying levels of detail. In this way, users can smoothly browse across the images in the database and focus on the relevant ones.

The results of the experiments show that this method is a significant improvement over the query-by-example method of almost 300%. The technique is suitable for scaling both the database and the similarity pyramid. Further, the color spaces Lab and RGB are both suitable for the perceptually uniform representation of images in this application. In conclusion, this new retrieval technique offers the user a good opportunity to focus on the desired images.

1. Introduction

In the past decade, there has been an explosion in the amount and complexity of digital data being generated, stored, transmitted, analyzed, and accessed. This is partially due to the advances in computer technology and the World Wide Web. Much of this information is multimedia in nature, including digital images, video, audio, graphics, and text data. Further, in different areas such as art history, medicine, advertising, entertainment, and industry, more and more image database are being made and used. In this paper, we focus on image pictorial information (Internet and image databases).

Traditionally, the search for images in a database is done through text-annotations. In this approach, the images are first annotated manually by keywords and then retrieved by their associated annotations. However, there are many problems with text-based image retrieval systems. It involves a large amount of manual effort, so it can take a long time to complete the search. Further, an image annotation can have several meanings, which can result in differences in interpretation of the image content. Also, there might be inconsistency of the keyword assignments among various people. Consequently, the resulting annotations might not be rich enough or consistent enough for different sorts of queries.

The goal of this paper is to introduce a user-friendly composite image retrieval system, where users can find images of a broad undefined category of images. The description of the image category can be a general one. In particular, we want users to be able to search for: scene categories and genre categories. Images in the same scene category present a similar situation, like soccer, sunsets, forests. Images in the same genre category induce the same emotion. It is assumed that scenes and genres can be characterized by color information. Therefore, for this application, the problem is reduced to finding the right color characteristics top capture different scenes or genres.

This method is designed according to the following criteria:

- The *initial query formulation* must be at a semantically high level. This means that the user can easily express a query on a semantic level. An example is o let the initial query be an image, in stead of a specification list of low-level features.

- A compact and hierarchical *overview* is needed of

the images in the database, especially if the image database is large.

- The user must be allowed to *refine the initial query* and search again.

To satisfy the three criteria, we aim at combining the most well-known and widely used techniques today for content-based retrieval; query-by-example, browsing and relevance feedback. All three have certain advantages and disadvantages. The three methods are complementary to each other, each providing a necessary part of our criteria. By making this combination, we expect to tackle the problem of the discrepancy between low-level image features and high-level concepts.

The *query-by-example* technique is suitable for the initial query formulation. The technique allows for a query to consist of an example image. The advantage is that the user doesn't have to specify the query in terms of low-level features, but can point out one or more example images. After image retrieval, *browsing* allows the user to move efficiently through the retrieved images to locate the desirable ones. The purpose of this technique is to get to a quick overview of the retrieved images. *Relevance feedback* is a technique which allows for several search loops in which the user can refine its query. This technique helps the user to refine the initial query and converge to the desired images in a sequence of steps by interaction.

We will apply the retrieval method on photographical images containing a wide variety of outdoor images, including images of nature (wildlife and landscapes). The purpose is to find similar groups of images in the database and not an exact match. In our experiments, we will focus on the Corel Stock Photo image database.

This paper is organized as follows. In Section 2, related work is discussed. An overview of the image retrieval system is presented in Section 3. Experimental results are discussed in Section 4.

2. Related Work

Recently, the search for images is being done on the Internet. Many new enterprises on the Internet, like Lycos and Yahoo!, specialize in retrieving information from the web. This information consists for a large part of visual information, which creates a growing need for efficient retrieval techniques for images. Today, as a consequence, many search systems are based on visual content [4, 8, 13]. Unfortunately, images and user queries can not be specified in semantic terms (e.g. "skiers in the snow"). Also, low-level image features, extracted from the images, are often too restricted to describe images on a semantic level [9]. This results in a discrepancy between syntactics and semantics. Some systems have paid attention to this problem [3, 11, 2].

Cox *et al.* [3] developed PicHunter, a retrieval system designed to test a novel approach based on a Bayesian framework to relevance feedback. PicHunter is designed to have a minimal, "queryless" user interface, so that its performance reflects only the performance of the relevance feedback algorithm. In each iteration of the search, the N most probable data are shown to the user, and the user's response is then used to update the probability distribution. At any given time during a search, four images are displayed on the screen. The user can select one or more images by clicking on them with the mouse. In this way, relevance feedback can only be performed by a selection out of four images.

Santini *et al.* [11] define an interface and an interaction model that mixes browsing and searching, that allows users to explore the image space in search of meaningful images. Rather than choosing individual images, the user can choose a whole region as interesting and the user can rearrange them directly into the display, dragging similar images close together and dissimilar images apart. They developed an exploratory interface for their image database El nino.

The objective of Chen et al [1] is to present hierarchical algorithms for efficiently organizing and searching an image database, particularly when they become large (¿10.000). They present the following tools for managing large image databases: 1) A fast search algorithm which can perform exact search. 2) A hierarchical browsing environment, which is called a similarity pyramid, which efficiently organized databases so that similar images are located nearby each other. In this way, the database is organized logically and predictably so that users can find the images that they need.

We believe that the user is benefited by a browsing environment with a wide choice of images to select as relevant. For this reason, we implement a browsing environment which will allow the user to keep an overview and select multiple images as relevant. For this reason, this image retrieval system is based on the work by Chen *et al* [1]. We integrate this browsing environment for the application of finding broad categories of images. Chen et al. concentrated on finding a fast search and clustering algorithms. Our goal is to see if this method is suitable for our application. Besides, we adapted the similarity pyramid according to our needs.

Verschil: B: pruning and reorganization, wij niet. B: By moving up or down the pyramid structure, the user can either zoom out, to see large variations in the database content, or zoom in, to investigate specific areas of interest. Wij: Only four images are shown to the user at each level. The top of the pyramid is shown first and the user can then decide which similar images he/she is looking for and move up and down the tree.

3. Approach

The idea of this composite image retrieval method is as follows: initially, we start with the query-by-example technique. The advantage is that the user can formulate the query in the form of an example image. For this purpose, standard images are taken from the image database. The user can select some images for the first retrieval operation, which should be treated as a trial run only, to retrieve a few useful items from a given collection. Based on the extracted image features a search is done on the image data set.

The set of most similar images will be retrieved and displayed on screen in the form of a hierarchical organized relevance pyramid. Each level of the pyramid is organized so that similar images are near one another. This pyramid is referred to as relevance pyramid, based on the work by Chen *et al* [1]. The bottom level of the pyramid is filled with the K nearest images retrieved from the database and clustered in groups of four. Each cluster consists of images that are very similar. Then, the pyramid is filled with the most representative image of every cluster. In this way, at the top levels of the pyramid, each image is a representative example of all the images at lower levels, which are filled with roughly similar images. This two dimensional organization allows users to smoothly browse across the images in the database. The benefit of browsing in the relevance pyramid is that an ordering is present, so the user doesn't need to browse randomly to find images of interest.

The user is allowed to remove or add image from/to the set of desirable images, which is called the relevance set. The system reorders the data set, according to the relevance feedback, and returns a new pyramid to the user until the user is satisfied with the results. In this way, the user can refine the query in an intuitive way and search statements can be refined without exact knowledge of the database.

Let's say, a user is looking for sunny outdoor scenes. First, the user is confronted with a subset of images in the database on the left side of the *user interface*, from which the user can select an image to put in the relevant set on the right side of the interface. Then the image query is processed and a search is done in the image *database* on the basis of color. From the k-nearest images found, the *relevance pyramid* is formed. This pyramid is presented to the user in the user interface. Now, the user can browse through the relevance pyramid by moving up and down the pyramid by clicking on the images. In this way, new images that fit the target description (sunny outdoor scenes) can be forwarded to the relevance set. If the user is not satisfied with the retrieved result, a new relevance feedback loop can start. The user can form anew query in the relevance set and again a search is done.

4. Color Information

The choice of image features is an essential step in the development of a retrieval system. The nature and the quality of these features determine the practical use of the image retrieval system. We focus on the purpose of the application when users have a broad category of images in mind. The description of the image category can be a general one. In particular, we want users to be able to search for *scene* categories, and *genre* categories. Images in the same scene category present a similar situation, like soccer, sunsets, forewsts. Images in the same genre category induce the same emotion. It is assumed that scenes and genres can be characterized by color information [**?**]. Therefore, for this application, the problem is reduced to finding the right color characteristics to capture different scenes or genres. In general, color-based retrieval is similar to the problem of finding images whose color distribution is globally similar to that of the query image.

As an example, if users are searching for image sin the same *genre*, they may be searching for warm images, which typically contain the colors red and orange [cite??]. Or users may be searching for scary feeling images, which may contain the colors black and/or purple. Different types of *scenes* in images can also be characterized by their color schemes. For example, pictures of beaches contain mostly a region of sand (yellow) and a region of sea (blue). Also, pictures of nature of a certain season have characteristic color patterns, think of typical summer or fall scenes.

Color can be described by different color systems. According to inherent characteristics of human perception, a color system is mostly a three-dimensional space arranging the visually perceivable colors. This geometric ordering provides each color with a corresponding position in the space. The goal of a color system is to represent and classify colors objectively and to provide a way to express color differences.

Color systems have been developed for different purposes [5, 6]. With this large variety of color systems, the question arises which color system to use for this application. Assuming the samples have been recorded under the same imaging conditions (i.e. camera, illumination, and sample pose), a perceptual uniform color system is most suitable [5].

A perceptual uniform color space facilitates the classification and the distance of colors as they are perceived. It has the objective of determining color difference by measuring the Euclidean distance, which separates color points in this space. The *Lab* color space developed by the Commission Internationale de L'Eclairage (CIE) in 1976 is such a perceptual uniform color space. The Lab system is based on the opponent color theory, computed form the XYZ color system, and is the most widely used system [5]. The sys-

tem has the property that the perceived color difference of two colors correspond to the Euclidean distance between the two colors in the color system. This property holds almost everywhere in this space, excepts for some hues (red exhibits a larger range than green and blue; colors close to gray are poorly discriminated). In conclusion, close colors in this space correspond to perceptually close colors. Therefore, we will model color in the Lab color space.

5. Method

5.1. Image Representation

After image feature extraction, a representation scheme is needed. The goal is to represent image in a n-dimensional feature space, where each image is captured in vector form. The vector space is made of n different color axes. Further, when retrieving images, some colors are more important than others and can therefore have different weights assigned to them.

Images are represented by their *color histogram*. A color histogram can be seen as a vector ... in a n-dimensional feature vector space, where each element h represents the number of pixels of the image having the color j labeled by its color coordinates. In fact, a color histogram counts the number of colors occurring in the image. It has been shown that color histogram are invariant to translation and rotation about the viewing axis, and change only slowly under change of angle of view, change in scale, and a large amount of occlusion [12].

Further, attention has to be paid to the selection of the similarity function used for the comparison of histograms derived from the images. The color histogram of the query image will be compared to the color histogram extracted from the images in the database. The similarity function is chosen according to the following criteria: highly discriminative power, metric. This results in the following formula [**?**]:

formule Salton and Buckley

, where N is the number of images in the database and n denotes the number of images to which a feature value is assigned. In this way, features are emphasized having high feature frequencies but low overall collection frequencies.

5.2. Search

To apply similarity queries on feature vectors, it is required to develop an index structure,l which supports nearest neighbor queries efficiently. In this way, the set of images similar to a particular query image, is retrieved by searching for feature vectors close to that of the given image. Existing multi-dimensional indexing techniques include Bucketing algorithm, k-d tree, priority k-d tree, K-D-B tree, hB-tree, R-tree and its variants R+-tree and R-tree.

One of the most promising approaches is the R-tree and its numerous variants (the SR-tree, the R+-tree, the R-tree, etc.). Unfortunately, the complexity of most search algorithms grows exponentially with the dimension of the vector space, making them impractical for dimensions above 15. Such high dimensionality is present in this image retrieval system. It has been shown that the SR-tree outperforms the R-tree in the context of high dimensionality [7]. The k-nearest neighbors are found by using an efficient computational algorithm. To that end, the image feature space is indexed by the SR-tree.

5.3. Clustering

After the k-nearest images of the query image are found, they are clustered in a three dimensional pyramid structure. The lowest level of the pyramid consists of all the k-nearest images clustered in groups of four. At every higher level, the image that represents a cluster is presented. In this way, most similar images are at the top of the pyramid. The goal here is to offer the user the capability to browse through the images and decide which images are relevant and which are not. Images that are close together according to their color and similarity measure, need to be close together in the similarity pyramid. To accomplish this, a clustering method is used. Bottom-up (agglomerative) clustering seems to offer superior performance for browsing application [1].

The relevance pyramid consists of a quad-tree based generated by the agglomerative clustering algorithm. Agglomerative clustering is based on a proximity matrix that contains the distances between all pairs of images in a collection. We start with each data point forming its own cluster and gradually merge clusters until all points have been gathered together in one big cluster. In the beginning of the process, the clusters are very small and very pure - the members of each cluster are few, but very closely related. Towards the end of the process, the clusters are large and less well defined. In other words, for a collection of N images, N-1 fusions take place to result in a hierarchic classification in which small clusters of closely related images are nested within larger and larger clusters of less related images.

Herschrijven: The creation of the proximity matrix is the first step of the agglomeration algorithm and works as follows. The proximity matrix is a table of all the pair-wise distances between points. At the beginning of the process there are N rows in the table, one for each record. Next, we find the smallest value in the priximity matrix. This identifies the two clusters that are most similar to one another. We merge these two clusters and update the proximity matrix by replacing the two rows that described the parent cluster with a new row that describes that distance between the merged

cluster and the remaining cluster. There are now *N-1* cluster and *N-1* rows in the proximity matrix. We repeat the merge step *N-1* times, after which all images belong to the same large cluster.

The distance between clusters is measured according to the *complete linkage* approach. In this method, the distance between two cluster is given by the distance between their most distant members. This method produces clusters with the property that all memebers lie within some known maximum distance of one another. Complete link clustering is dilating, and therefore tends to create balanced trees [1].

The mapping of the quad-tree to the pyramid is performed starting at the leafs of the quad-tree and moving to its root. The leafs are formed out of the original k-nearest neighbor images found in the search. For example, if 128 nearest neighbor iamges are found, there will be 128 leafs in the quad-tree. For every four images in the clusters at the lowest level of the quad-tree, the most representative image is taken to fill the next level up the pyramid.

We use that histogram-based distance measure between images. We use the same distance measure to decide which image will be bubbled up to form the relevance pyramid. We do this by calculating the median of the leafs attachted to the emply node.

5.4. Relevance Feedback

If the user is not satisfied with the retrieved result, the desired images can be selected by the user and put in the relevance set. Together these images are the new query formulation. The reformulated query can then be compared with the images in the database in a new search operation.

The aim of relevance feedback is to construct new queries with a greater degree of similarity with the images previously identified as relevant than the original queries. At the same time, the new queries are expected to be less similar to the images identified as non-relevant. The assumption is that the reformulated queries will retrieve more images resembling the relevant ones previously retrieved, and fewer images resembling the non-relevant ones [10]. An effective feedback query following the retrieval of *n* relevant images can be formulated as:

formule salton88.

where I represent iamge vectors, —I— is the corresponding Euclidean vector length, and *n* is the number of relevant images in the collection. Q and Q denote the image feature vectors constructed for initial and the first iteration image example queries. The summation is taken over the relevant images.

6. Experiments

We present experimental results to test the retrieval performance of our method, using databases of 5.000 images containing a wide variety of photographical outdoor images including images of nature (wildlife and landscapes). In these experiments we try to show that the browsing opportunity of the similarity pyramid in combination with the relevance feedback method allows the user to focus on the preferred images. The preferred images are of the same scene or genre category, as defined by an individual user.

Three types of searches for the testing of the retrieval performance are distinguished [3].

- *Target search.* Users try to find specific target images.
- *Scene by Category Search.* Users seek one or more images from general categories such as: sunsets, forests, soccer. This is more complex than target search because it places more emphasis on the semantic content of images.
- *Genre by Category Search.* Users seek one or more images from categories which induce a certain emotion, such as: cold, scary, harmonious. This is even more complex than scene by category search because the content can only semantically be described and requires subjective judgments.

Doing experiments for each search type is important, but it is hard to find good measurements for the genre and scene searches, as it is hard to define those categories. For target search there are no such problems and several clear measures of effectiveness exist. Though target testing is directly analogous only to target search applications, systems which perform well on target testing are likely to perform well for scene and genre category search as well [3]. To this end, target testing is a useful paradigm for quantifying the effectiveness of our image retrieval system.

6.1. User Profile

To measure the performance of the retrieval method, we set the following user profile. We assume that 4 relevance feedback loops is the most a user would be willing to do. This implies that the system is unsuccessful if the target image is not found within 5 retrieval sessions. Another assumption is that the average number of user interactions in the similarity pyramid is at most 15 mouse clicks. Also, the average time it takes to find the target image within each feedback session is approximately 30 seconds. These assumptions are derived from various trial sessions.

Relevance feedback is implemented in such a way that users can choose up to 6 images in the relevance set. These new images in the relevance set form the new query. The

new query features are calculated by taking the mean of the features of the image sin the relevance set. The relevance set can contain the original example image, but can also contain newly found images in the similarity pyramid.

7. Conclusions

In this paper, the goal is to introduce a new method of image retrieval. We combine the following three techniques for image search; query-by-example, browsing, and relevance feedback. The purpose of the application is finding images of the same scene or genre. After doing various experiments, we come to the conclusion that this method is effective for the purpose of this application. In general, it improves the query-by-example method with almost 300%. The technique is suitable for scaling both the database and the similarity pyramid. Further, the color spaces Lab and RGB are both suitable for the perceptually uniform representation of images in this application. In conclusion, this new retrieval technique offers the user a good opportunity to focus on the desired images.

References

[1] J.-Y. Chen, C. A. Bouman, and J. Dalton. Hierarchical browsing and search of large image databases. In *IEEE Transactions on Image Processing*, 1999.

[2] J. Corridoni, A. DelBimbo, and P. Pala. Image retrieval by color semantics. In *Multimedia systems*, volume 7, pages 175–183, 1999.

[3] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Target testing and the pichunter baysian multimedia retrieval system. Washington D.C., May 1996.

[4] M. F. et al. Query by image and video content: the qbic system. In *IEEE Computer*, volume 28, 1995.

[5] T. Gevers. Color in image databases. Amsterdam, The Netherlands, 1999.

[6] T. Gevers and A. W. Smeulders. Color-based object recognition. In *Pattern Recognition*, volume 32, pages 453–465, Mar. 1999.

[7] N. Katayama and S. Shin'ichi. The sr-tree: An index structure for high-dimensional nearest neighbor queries. In *ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, 1997.

[8] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. In *SPIE, Storage and Retrieval Image and Video Databases II*, volume 2185, San Jose, CA, 1994.

[9] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. In *IEEE Transactions on Circuits and Video Technology*, Urbana, IL.

[10] G. Salton and C. Buckley. *Automatic Text Retrieval*. New York, USA, 1988.

[11] G. Salton and C. Buckley. In automatic text retrieval. 1988.

[12] S. Santini and R. Jain. Interfaces for emergent semantics in multimedia databases. In *Proceedings of SPIE/IS&I Conference on Storage and Retrieval for Image and Video Databases VII*, volume 3656, San Jose, CA, 1999.

[13] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, May 1991.

[14] J. Vendrig, M. Worring, and A. Smeulders. Filter image browsing: interactive image retrieval by using database overviews. Technical report, Faculty of Science, University of Amsterdam, The Netherlands, 1998.