

Multiresolution similarity search in image databases

Martin Heczko¹, Alexander Hinneburg¹, Daniel Keim², Markus Wawryniuk²

¹ Institute of Computer Science, University of Halle, 06099 Halle (Saale), Germany
email: {heczko,hinneburg}@informatik.uni-halle.de

² Department of Computer & Information Science, University of Konstanz, 78457 Konstanz, Germany
email: {keim,wawryniu}@informatik.uni-konstanz.de

Abstract. Typically searching image collections is based on features of the images. In most cases the features are based on the color histogram of the images. Similarity search based on color histograms is very efficient, but the quality of the search results is often rather poor. One of the reasons is that histogram-based systems only support a specific form of global similarity using the whole histogram as one vector. But there is more information in a histogram than the distribution of colors. This paper has two contributions: (1) a new generalized similarity search method based on a wavelet transformation of the color histograms and (2) a new effectiveness measure for image similarity search. Our generalized similarity search method has been developed to allow the user to search for images with similarities on arbitrary detail levels of the color histogram. We show that our new approach is more general and more effective than previous approaches while retaining a competitive performance.

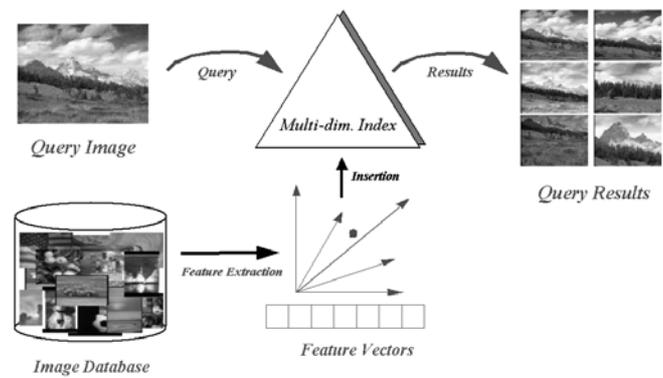


Fig. 1. The concept of feature-based image similarity search

1 Introduction

Among the rapidly increasing amount of information stored in today's computer systems, images play an increasingly important role. People ask for systems allowing them to store, manage, and retrieve images with good effectivity and efficiency. The task of so-called *image retrieval systems* is to find the most similar images for a given query, which can be an image or a sketch. Well-known examples are retrieval systems for the WWW [11,13,14], medical databases [37], or CAD databases [3]. Most commercial systems still use a text-based search based on captions and only rely on the textual information stored together with the images. More sophisticated systems use features of the images to determine the similarity with respect to the query image. The general process of a feature-based image similarity search is shown in Fig. 1. The feature vectors are extracted from the image database and inserted into a multidimensional index. The feature transformation is also applied to the query image, and the resulting feature vector is used to query the multidimensional index to obtain the query results. The similarity measure used in this approach is mainly defined by the feature vectors used.

Color-histogram-based feature vectors are among the most widely used feature vectors in image retrieval systems. Color histograms have the advantage that they contain important, highly aggregated information about the images and are easy and fast to compute, making them applicable to very large datasets. But similarity search with histograms based on a global notion of similarity has a limited effectiveness, as we show in this paper.

In this article, we focus on improving the widely used histogram-based image similarity search in order to overcome the limited effectiveness. We found that it is not enough to compare histograms as a whole but that a comparison of smaller subhistograms can improve the quality of similarity search considerably. Our idea is to define a hierarchy of similarity measures that allows the user to search on different resolutions.

Our article is organized as follows. Section 2 provides a brief survey on image similarity search. We outline the contribution of our work in Sect. 3. In Sect. 4 we describe and formally define our new method, and in Sect. 5 we develop a new effectiveness measure. In Sect. 6, we present the efficiency and effectiveness results.

2 Image similarity search

The general process of a feature-based image similarity search is shown in Fig. 1. The feature vectors are extracted from the image database and inserted into a multidimensional index. The feature transformation is also applied to the query image, and the resulting feature vector is used to query the multidimensional index to obtain the query results. The similarity measure used in this approach is mainly defined by the feature vectors used. Color-histogram-based feature vectors are among the most widely used feature vectors in image retrieval systems. Color histograms have the advantage that they contain important, highly aggregated information about the images and are easy and fast to compute, making them applicable to very large databases.

The important question “What is similar?” remains unanswered, from both the user’s and the computer’s point of view. This makes it difficult to design an image retrieval system that works well in a large range of applications. The reason is twofold: first, there are many subjective opinions about “similar” and “not similar”. One person sets a high value on some specific characteristics, whereas another person probably sets a high value on other, i.e., opposite or complementary, characteristics. Conferring this ambivalence into algorithms is not straightforward. Relevance feedback is a promising approach. The situation will get better in a well-defined application domain, i.e., a surgeon should have some established criteria to find similar images showing tumors. The second reason is closely related to the very nature of similarity search. Searching large databases for similar images is usually based on extracting and comparing certain features of the images. The fundamental idea is that similar images have similar features, i.e., the feature vectors have a small distance with respect to a given metric. Unfortunately, similar images do not necessarily imply similar feature vectors and vice versa. As an example, consider the color histograms describing the distribution of colors in an image. Figures 2a and 2b show similar images. Objects in these images contain similar proportions in certain color ranges, but the corresponding feature vectors are different.

2.1 Overview of feature-based image similarity search

Searching large databases for similar images is usually based on extracting and comparing certain features of the images. More precisely, image retrieval or image similarity search is done as follows (Fig. 1).

First, predefined characteristics (also called features) are extracted from the query image, resulting in a so-called feature vector. Image scientists have designed a large number of features to find a meaningful mathematical representation of important characteristics of an image. Meaningful feature vectors are an important topic throughout the literature. To overcome different understandings of similarity and different retrieval strategies, methods of relevance feedback are applied, allowing the user to refine the proposed similar images and to concentrate on some specific characteristics.

Second, the database, which contains the feature vectors for all images stored, is searched for feature vectors with a small distance to the query feature vector. This is done on the

assumption that similar feature vectors (with respect to the used metric) imply similar images. Database researchers have developed efficient index structures for the nearest-neighbor search problem to improve efficiency.

Clearly, the first step directly affects the effectiveness, whereas the second step directly affects the efficiency of the image retrieval system. Further on, these two measures interact with each other [17]. The longer the feature representation is, the better the quality of the retrieval gets, but the larger the execution costs become. In other words, an improvement in effectiveness leads to a deterioration of performance and vice versa.

The significance of effectiveness and efficiency changes during the process of searching similar images. From a user’s perspective, searching for images typically involves several steps. In the first few steps, a user refines his or her query with the help of relevance feedback until the matches are sufficiently good. In the final step of the search process, the archive is extensively searched for all relevant images. Obviously, retrieval effectiveness in the first few steps is not as important as retrieval efficiency. In the final step, on the other hand, quality plays the key role and a user is ready to tolerate longer response times if more relevant images are retrieved.

One crucial task of image similarity search is the extraction of feature vectors. The used features directly affect the effectiveness. Not every feature is appropriate for every application domain, and, conversely, for a particular application domain only certain specific features are useful. Examples of features are the color distribution of the pixels in images [9,31], the shape of objects in images [18,19], the spatial arrangement of color sets [5], the texture of images [24,40,41], the spatial correlation of colors [16], the degree to which pixels of a color are members of large similarly-colored regions [22], attributes of image regions [10,21], etc. There are several possibilities proposed in the literature to involve features in similarity search. For instance, the feature vector is computed for the overall image, the image is divided into regions and the feature vector is computed for each region, or the search is done with a combination of features.

A number of image retrieval systems have been built that support the features mentioned above. Many systems are based on color histograms [7,32,34], others support combinations of features [7,25,26,32,34]. For example, QBIC [1] combines color histogram, shape, and texture features. Even others try to do some partial matching of images, for example WALRUS [21]. The techniques used in image retrieval are taken from a number of different areas including pattern matching [30], information retrieval [2], and computer graphics [28]. A number of techniques have been developed to speed up the search process. The developed techniques range from advanced high-dimensional indexing techniques [4] over fast linear scans of a compressed version of the feature vectors [38] to parallelization techniques [39].

Techniques based on wavelets are used by [28,34,35]. These approaches basically apply an image wavelet transformation and use specific wavelet coefficients to compute the feature vector. A discussion of the differences between the Haar and Daubechies wavelet transformation can be found in [35].

2.2 Image similarity search with histograms

In most cases, statistical information about the images serves as a basis for a similarity measure. This information is usually given by some form of a histogram, and thus most similarity search systems are somehow histogram based. In this paper, we therefore focus on histogram-based techniques, especially color histogram techniques. Color histograms have a number of advantages: they contain important information about the images and are easy and fast to compute, making them applicable to very large databases. An unsolved problem, however, is the limited effectiveness of current histogram-based image retrieval systems. For better retrieval quality a notion of similarity is needed that is more general than the simple (usually global) color-based similarity used in most existing systems.

Other scientists have also suggested solutions to improve the quality of histogram-based techniques. In [8] an image is divided into regions with homogeneous color distribution, and for each region a histogram is computed, resulting in a histogram family. To determine the similarity between the query image and a database image, corresponding regions must be identified. The matching of the histograms and areas occupied flows into the computation of the similarity value. Another approach is presented in [23]. The authors point out that simple color histogram techniques do not reflect the fact that, despite similar histograms, i.e., similar color distribution, images can look completely different because the location of pixels with the same color is not taken into account. For this reason not only are the pixel's colors processed but also the pixel's edge density, texturedness, gradient magnitude, or rank can be regarded, resulting in a multidimensional histogram.

Typically, histograms are compared bin by bin and the differences are added up somehow. This does not reflect the fact that neighboring bins represent a higher similarity than distant bins. It is possible that images with similar colors may have the same distance, as opposed to images with completely different colors. The quadratic form [12,26] considers the similarity of bins (colors) by incorporating a matrix denoting the similarity between bins (colors).

Other papers, such as [6,29], deal with a formal analysis of histograms and their limitations. The authors of [29] discuss the question of how many distinguishable histograms can be stored (capacity) and how the average number of returned images depends on the retrieval threshold (sensitivity). This is done to enable the user to test the performance of color histogram indexing via processing a small sample of images.

3 Our contribution

To exemplify the potential of our ideas, consider the following images. Figure 2 shows two examples of similar images. The two pictures in Fig. 2a basically show the same object with the second image showing the object much closer. The corresponding histograms show some similarities but also clearly indicate the higher frequencies in the darker range for the second image because of the relatively larger telephone. There are similarities of the histogram shapes, but most existing image similarity systems would not be able to discover them due to the overall differences. This becomes even more obvious in the second example. Figure 2b shows three images that are

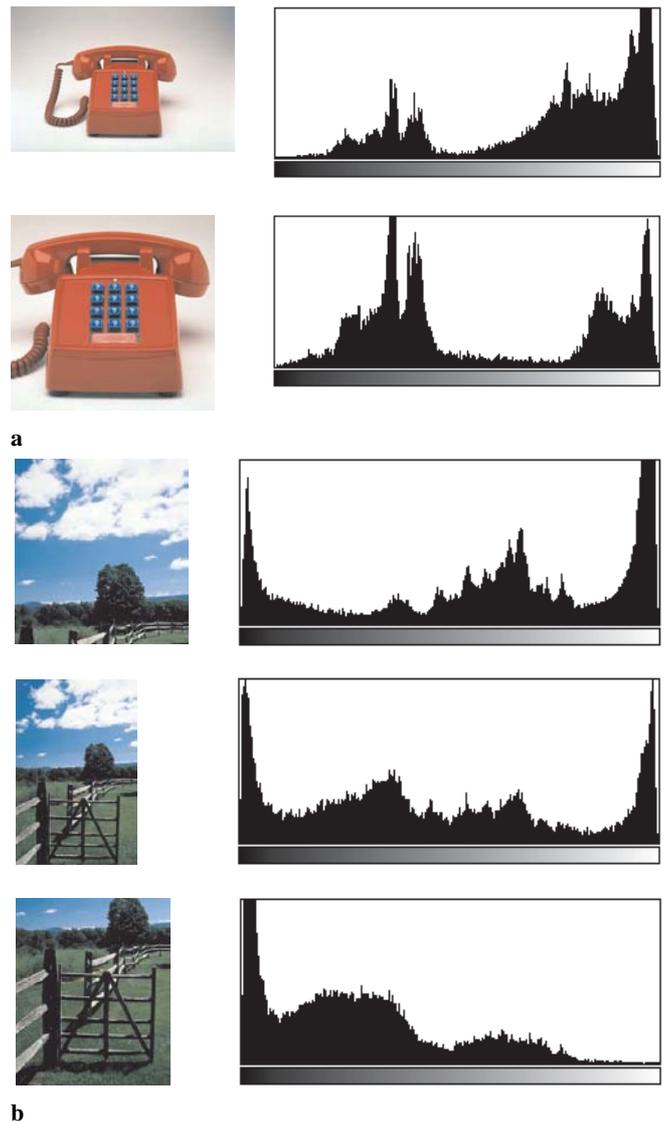


Fig. 2a,b. Similar images and their corresponding luminance histograms. The histograms have similar subhistograms, which correspond to the similarity of the images, but common measures of similarity will not detect the similarity. **a** Telephone. **b** Countryside

clippings of the same picture. Although there is a clear similarity in the images, the corresponding color histograms seem to be quite different. When considering them more closely, however, one may discover some similarities. It is clear that the similarity of the images is still hidden in the histograms, but, due to the differences, it cannot be found by a standard histogram-based similarity search. The color histogram contains important information, i.e., the similar proportions in specific color ranges, which can be used to determine the similarity of the images, regardless of different sizes resp. different relative frequencies in the histograms.

The question is how to discover those similarities automatically. It is not enough to compare histograms as a whole; in fact, comparisons of smaller subhistograms can improve the quality of similarity search considerably. Our basic idea is to divide the histograms into a number of subhistograms

containing the relative frequencies of certain color nuances. In general, the range of color nuances of an object is well separated and the distribution of color nuances is uniform, which allows us to find similar images in cases as described above and enables even new similarity measures. In addition to the subdivision of the histograms, we have to apply a feature transformation such as a simple normalization or a wavelet transformation. Each subdivision together with the chosen transformation results in a different notion of similarity. In other words, our basic idea is to partition the histograms and to apply a feature transformation to the subhistograms. This leads to our generalized notion of similarity.

Besides the subdivision of the histograms we show how to define a hierarchy of similarity measures that allows the user to search on different resolutions. We propose a multi-level approach: histograms are divided in a hierarchical way, and the user can select a specific level for comparison. We implemented our ideas using a wavelet transformation of the color histograms. Wavelet theory provides a nice framework for a hierarchical decomposition of the color histograms. In querying the database, any similarity measure defined in the hierarchy of similarity measures can be used separately or in combination.

In the context of this article, we aim at similarity search on color histograms. Our contribution is a generalization of the similarity measure that considers histograms on various detail levels. The major difference with other approaches using histograms is that we present a new similarity measure for histograms, whereas other work aimed at new features using histograms.

The experiments show that even for a simple (i.e., global) similarity measure, our technique is more effective than existing approaches such as the WBIIS or WISE systems [32, 35, 36]. We also show that our technique is more general than existing approaches and allows one to find images that are classified as being similar by a human, although their histograms are rather different. In this context, we also propose a new effectiveness measure that, in contrast to the well-known precision and recall measures, is independent of the size of the result set and takes the ordering of the returned images into account. A performance evaluation shows that our system provides a competitive performance.

4 Histogram-based image similarity

In this section, we give a formal description of our generalized histogram-based similarity measure. We introduce the general idea and describe how the generalized similarity can be implemented using wavelets.

4.1 Overview

A histogram characterizes the distribution of samples. There are mainly two possibilities for building the color histograms of an image. First, each pixel of an image is taken as a sample, resulting in a vector that gives the relative frequency for each color. Second, one can split the color channels of the color model used to get the samples and to form one histogram for each color channel consisting of the intensities of the particular

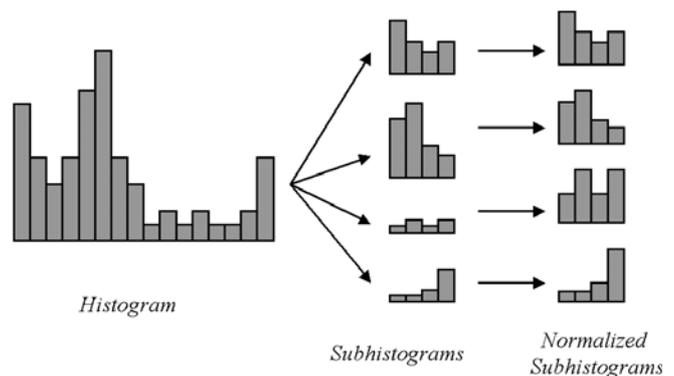


Fig. 3. Histogram with subdivision and normalization

channel. We have used the latter method to obtain histograms for different color models, namely, the RGB and HCL color models. For the RGB color model we get histograms for the red-green-blue channels, and for the HCL color model we get histograms for the hue-chromacity-luminance channels. The following considerations are made for one histogram, but the results can be used for the combination of the specific three histograms as well as for the RGB and HCL color models.

For an effective histogram-based similarity search, it is important not to just perform a piecewise comparison of the histogram vectors but to use more complex distance metrics as, for example, proposed in [26]. Even more effective are approaches that apply a feature transformation to the histograms [35]. The feature transformation is chosen to extract certain characteristics that will be used for a comparison instead of the original histograms.

Our results show that feature transformations are not enough to cover a number of obvious similarities. We therefore propose a subdivision of the histograms and apply feature transformations to them. This allows us to define a similarity measure that significantly improves the effectiveness.

4.2 Generalized histogram-based similarity

To define our generalized image similarity measure, we first need to give a formal definition of histograms. In the case of color histograms, the variable is one color channel. A histogram for one color channel with n bins is denoted by $c = (c_1, c_2, \dots, c_n)$ and the c_i ($1 \leq i \leq n$) are the relative frequencies. A subhistogram is a sequence of successive elements of c . For our purposes, a set of subhistograms should have the following properties: each c_i is covered by one subhistogram, and the subhistograms do not overlap each other. To model N subhistograms s_j ($1 \leq j \leq N$) of the original histogram c , we use a set of subdivision points $T = \{t_1, t_2, \dots, t_N\}$ with the property that $1 = t_1 < t_2 < \dots < t_N < n$. The subhistogram s_j starts at t_j and ends just before the next subdivision point t_{j+1} (or n if $j = N$): $s_j = (c_{t_j}, \dots, c_{t_{j+1}-1})$ with $t_{N+1} = n + 1$. The entire set of subhistograms s_j of c is denoted by $S(T, c) = (s_1, \dots, s_{|T|})$. Note that we do not introduce an index to access the individual elements of s_j because we only refer to the entire s_j . Figure 3 shows a histogram and subhistograms for $n = 16$, $N = 4$, and $T = \{1, 4, 8, 12\}$.

Next, we define a similarity measure for subhistograms. As in the case of normal histogram-based similarity, we first apply an appropriate feature transformation f to the subhistograms s_j . This feature transformation can be a simple normalization (Fig. 3) or a complex transformation such as a wavelet transformation. The similarity of two subhistograms $s_j \in S(T, c)$ and $s'_j \in S(T, c')$ can then be determined with a distance metric δ : the smaller $\delta(f(s_j), f(s'_j))$ is, the more similar are the subhistograms. Given a subdivision T , a feature transformation f , and the distance metric δ , we define the similarity A of two histograms c and c' by

$$A(T, f, \delta) = \sum_{j=1}^{|T|} \delta(f(s_j), f(s'_j)).$$

By introducing a subdivision of the histograms we are able to describe a more general similarity measure that allows us to focus the search on the important portions of the color distribution, corresponding to the characteristic objects in the image. If the subdivision, the distance metric, and the feature transformation are chosen appropriately, then we can find similarities as described in Sect. 3.

4.3 Extension to a hierarchy of similarities

Our generalized similarity allows a specification of a hierarchy of subdivisions that easily extends to a hierarchy of “coarser” and “finer” similarity measures independently of the feature transformation. The definition of a similarity measure using subdivisions of histograms allows the user to go from a global comparison of the color distribution to a more local comparison. Adding subdivision points to an existing histogram results in a similarity measure’s relying upon finer color nuances. The finer the subdivision becomes, the less relevant is the global color distribution and the more influential are the local properties of the color distribution. A hierarchy of similarities

$$A^0(T^0, f^0, \delta), A^1(T^1, f^1, \delta), \dots, A^l(T^l, f^l, \delta)$$

can be obtained by a sequence of subdivisions $\emptyset = T^0 \subset T^1 \subset T^2 \subset \dots \subset T^l$ and the corresponding feature transformations f^k ($k = 1, \dots, l$). Using all subdivisions and corresponding similarity measures allows a search for “finer” or “coarser” similarities as well as combinations of them. But what is a good hierarchy of subdivisions and how can the corresponding similarity measures be calculated easily? The next subsection answers this question.

4.4 Wavelet-based instantiation

There are a number of possibilities to instantiate our generalized similarity measure. An instantiation just needs to define the strategy of choosing the subdivision points and the corresponding feature transformations. In this subsection, we present a wavelet-based solution that works efficiently and effectively and provides new potentials for image similarity search. By using the multiresolution properties of the Haar

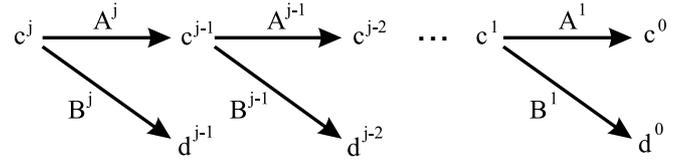


Fig. 4. Schema of the wavelet transformation

wavelet transformation, we naturally get a hierarchy of subdivisions and similarity measures as described in the last subsection. More details on the wavelet theory and the multiresolution analysis can be found in [28].

Our hierarchy of subdivisions $\emptyset = T^0 \subset T^1 \subset T^2 \subset \dots$ of a histogram c with n bins ($n = 2^r$) is the sequence of subdivisions T^k ($k = 0, \dots, r-1$) defined as the ordered set

$$T^k = \bigcup_{j=0}^{2^k-1} \left\{ j \cdot \frac{n}{2^k} + 1 \right\} \quad (1)$$

The j -th subhistogram at the k -th level of the hierarchy is denoted by s_j^k . Note that all subhistograms s_j^k ($k = 0, \dots, r-1, 1 \leq j \leq |T^k|$) from the set of subhistograms $S(T^k, c)$ generated by the subdivision T^k have the same size of 2^{r-k} bins, whereas the original histogram c is divided into 2^k subhistograms.

Now we have to apply the feature transformation f^k on the subhistograms s_j^k (corresponding to the subdivision T^k) according to the definition of our generalized similarity. The Haar wavelet transformation recursively applies the averaging matrices A^m and the differencing matrices B^m according to Fig. 4. Our feature transformation $f^k : \mathbb{R}^{r-k} \rightarrow \mathbb{R}$ calculates the coarsest detail coefficient c^0 when s_j^k is transformed:

$$f^k(s_j^k) = \begin{cases} B^1 \cdot A^2 \cdot \dots \cdot A^{r-k} s_j^k & k < r-1 \\ B^1 \cdot s_j^k & k = r-1. \end{cases} \quad (2)$$

The subhistograms of subdivision T^{r-1} only consist of two values (bins). The transformation only applies matrix B^1 to compute the coarser detail coefficient for each subhistogram. The next coarser subdivision T^{r-2} obtains subhistograms with four values that will be transformed with $B^1 \cdot A^2$. The entire Haar wavelet-transformed representation of the four values of one subhistogram would consist of two finer detail coefficients and one coarser detail coefficients and the overall average. The finer detail coefficients are exactly the coefficients calculated for the subhistograms of the finer subdivision T^{r-1} . The same can be applied recursively on the coarser subdivisions. This leads to the following statement: all feature transformations of the subhistograms resulting from subdivision T^k contain the same information as the detail coefficients d^k of the wavelet-transformed representation of c (Fig. 4). Therefore, we have

$$\begin{aligned} d^k &= B^{k+1} \cdot A^{k+2} \cdot A^{k+3} \cdot \dots \cdot A^r \cdot c \\ &= (f^k(s_1^k), f^k(s_2^k), \dots, f^k(s_{2^k}^k)). \end{aligned}$$

As a result, we can apply the Haar wavelet transformation to the histogram c and we get all feature-transformed subhistograms together with the wavelet-transformed representation

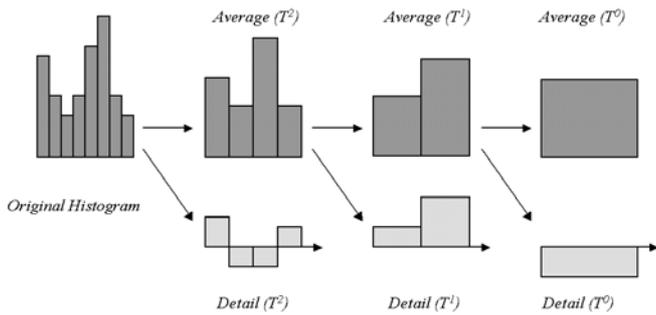


Fig. 5. Example of a wavelet transformation of a histogram

$(c^0, d^0, d^1, \dots, d^{r-1})$. All subdivisions and corresponding feature transformations are done in one step. An example wavelet transformation of a histogram is shown in Fig. 5.

Given the sequence of subdivisions $T^0 \subset T^1 \subset \dots \subset T^{r-1}$ as defined in Eq. 1 and the corresponding feature transformations f^k as defined in Eq. 2, our hierarchy of similarity measures A^k ($i = 0, \dots, r-1$) can now be defined as

$$A^k(T^k, f^k, \delta) = \sum_{j=1}^{2^k} \delta(f_j^k(s_j^k), f_j^k(s_j^{k'})). \quad (3)$$

Note that each similarity measure defined by our hierarchy of similarity measures in Eq. 3 is based on just one subdivision level. But all subhistogram similarities equally contribute to the overall similarity. To allow a flexible search, the similarities defined by different subdivision levels may be combined and weighted. In the next subsection we discuss this idea.

4.5 Combining detail levels

One advantage of our approach is that it combines the single detail level coefficients d^0, \dots, d^{r-1} into a new general similarity measure. The combination of multiple detail level coefficients allows a flexible search, focusing on certain details while still preserving the global context. There are many possibilities to combine the detail level coefficients.

The general approach behind the idea of combining detail levels is an arbitrary weighting of the subhistograms on different detail levels. The weighting allows the user to focus on certain ranges j of the histograms and arbitrarily combine different detail levels k . Let T be the vector of subdivisions (T^0, \dots, T^{r-1}) and f the vector of the corresponding feature transformations (f^0, \dots, f^{r-1}) . Then, our extended similarity measure A can be formally defined as

$$A(T, f, \delta) = \sum_{k=0}^{r-1} \sum_{j=1}^{2^k} w_{kj} \delta(f_j^k(s_j^k), f_j^k(s_j^{k'})), \quad (4)$$

where w_{kj} are user-provided weights. This allows a very general similarity search, but there is no general method to determine the $2^r - 1$ weights in order to maximize the effectiveness (precision, recall). An appropriate way to do this is to restrict the weights, e.g., all weights corresponding to the same detail level have the same value – either 0 or 1. This leads the task

to choose a good combination of detail levels. In Sect. 6 we will come back to this question.

Focusing on the finer levels, the average coefficients are neglected, which can be interpreted as an abstraction from the absolute frequency of the colors. Instead, the similarity measure focuses on the differences of color frequencies. Figure 8 will justify this theoretical consideration. In most real applications, the difference of color frequencies directly corresponds to some characteristic structure in the images, which is the reason for the effectiveness of our approach.

5 Measures of effectiveness

The most important criterion for an evaluation of our approach is its effectiveness. But the effectiveness of an image similarity search system is hard to measure, and confirming the semantical correctness is difficult and subjective.

5.1 Recall and precision

Two well-known measures to determine the effectiveness are recall (R_r/R) and precision (R_r/E) (with R_r , R , and E as defined below). To rate the effectiveness of a system, the pairs of precision and recall (determined for a given number of queries) are calculated and plotted in a so-called recall-precision diagram. A system gives good performance if many points lie near the point (1, 1), i.e., recall and precision are near 1. The advantages of recall and precision are that they are well known and widely used, and that they give an overview of all results. The disadvantage of recall and precision is that they depend heavily on the number of returned images and do not account for their ordering. If more images are returned, then recall and precision are more likely to obtain more relevant images, but the precision will decrease. Returning the whole database has a guaranteed recall of 1.0, but the precision will be almost zero.

To compare different similarity search systems, one has to compare the different precision recall plots. First, this task is subjective, and second, this is not easy because there is a tradeoff between precision and recall. Hence it is difficult to compare the results shown in Fig. 6, where different systems return a fixed number of query results (especially if parameters like R , R_r , and E vary). To compare the effectiveness despite that problem, usually plots of the average precision with fixed values for the recalls are made. But the resulting curves are still hard to compare since one has to decide whether recall or precision is more important. Figure 6 shows the recall-precision plots of the results of several methods tested. Each point represents one test query. The results show that on average our approach provides a higher recall than the other approaches while the precision remains approximately the same.

5.2 A new effectiveness measure

As already mentioned, recall and precision are interdependent. In addition, the ranking of the returned relevant objects is not taken into account, which is also very important, even though in most cases similarity search systems provide a meaningful ranking of the results.

To overcome these problems, we introduce an effectiveness measure that allows a better comparison of the different approaches. To define this effectiveness measure, we need a number of data- and query-dependent parameters such as the number of relevant images in the database (R), the number of returned images (E), the number of returned relevant images (R_r), and the number of missed relevant images ($R_m = R - R_r$). This requires that for a sample query the complete set of correct similar images must be known. Typically this is done by selecting the query image and similar images by hand (Sect. 6). Recently the authors of [27] presented a nice idea to select the *relevant result set*, which contains images that are expected to be similar to the query image: For a given query image, the result sets of different search systems are combined. The result of a query is usually a list of images sorted in descending order of similarity. Our effectiveness measure takes the ranking of results into account and considers the missed relevant images. Essentially, the quality measure is defined as the ratio of the sum of ranks of all relevant images $SumR_{all}$ and the sum of optimal ranks $SumR_{Opt}$. Clearly, the optimal result is where all relevant images occupy the first ranks of the result list. This gives the value

$$SumR_{Opt} = 0 + 1 + \dots + (R - 1) = \frac{R(R - 1)}{2}.$$

As mentioned, $SumR$ is the sum of rankings over all returned relevant images. In order to consider the missed relevant images, too, we assign the ranks $E, (E + 1), \dots, (E + R_m - 1)$ to the R_m missing images, i.e., if a relevant image is not in the result list of the retrieval, we assign best case ranks to it, assuming that they are following right after all images of the result list. This is an optimistic but fair assumption. Due to the fact that the retrieval system retrieves only the top E images, there is no knowledge about the ranks of the missing images. Therefore, we assume the best ranks, i.e., right after the retrieved images. The sum of rankings including the missed relevant images $SumR_{all}$ can therefore be calculated as

$$SumR_{all} = SumR + E + \dots + (E + R_m - 1).$$

Now our effectiveness measure can be defined as

$$eff = \frac{SumR_{Opt}}{SumR_{all}}.$$

Obviously, the better the retrieval, the smaller $SumR$ and the greater eff becomes. But the range of eff depends on the number of returned images E and the number of relevant images R and is given by $[\frac{R-1}{2E+R-1}, 1]$. The minimum value of eff is the effectiveness measure where all relevant images are scored at positions $E, E + 1, \dots, E + R - 1$. Because R varies for each query, we normalize eff and finally obtain the normalized effectiveness measure EFF with a range of $[0, 1]$. The normalization enables us to combine the quality measures for different queries with different numbers of relevant images. Finally, the effectiveness of a feature is given by the average \overline{EFF} of the effectiveness values EFF for each query in the test, which provides insight into the quality of an algorithm. Using the same database, the same sample queries and the same value for the number E of objects to return, we

can compare several features and feature combinations based only on their \overline{EFF} values.

Recently [20] introduced an effectiveness measure that takes rank into account as well. In contrast to our method this effectiveness measure requires retrieving all relevant images from the database, i.e., in the worst case a scan over the full database is required. Our measure is applicable if the number of images to return is less than the size of the database and only this number is retrieved. Other ideas to evaluate the effectiveness that also use rank are discussed in [27].

6 Experimental results

In this section, we provide the results of our experimental evaluation and compare the effectiveness and efficiency of our approach to the effectiveness and efficiency of previous approaches. The test database, which can be downloaded at [33], contains about 10,000 color images.

The comparisons were done with online image search engines from the universities in Stanford [32] and Munich [25], which use the same image database. A prototype of the system proposed in this paper is available on the Web [15]. The effectiveness of all systems were tested with 32 sample queries with varying content. For each sample query the correct and complete set of similar images is known. Those images can be determined easily due to the organization of the image database used in our case. The sample queries used and the expected results can be found at [15]. All tested algorithms had to return $E = 20$ potential similar objects. The approaches used for comparison are

- **Adapt:** A system that uses quadratic form distance functions for similarity estimation [25,26] (several predefined similarity matrices can be used; the three best were used).
- **WBIS:** A similarity search based on a wavelet transformation [32,36].
- **Color Histogram:** A histogram similarity search which does not use a feature transformation [32].
- **Color Layout:** A layout-based approach taking the spatial distribution of colors into account [32].
- **Our Approach:** Using all Haar wavelet detail information of the transformed color channel histograms weighted with the number of coefficients of one detail level. The similarities for each color component are simply combined by adding them up. The L_1 -norm was used as distance metric δ .

The results of the comparison are shown in Figs. 6 and 7. As already mentioned in Sect. 5, Fig. 6 shows the recall-precision plots for the methods tested. Each point represents one test query. The plots show that, on average, our approach provides a higher recall than the other approaches while the precision remains approximately the same. This is confirmed by the average recall-precision values, where our method yields the best effectiveness in recall as well as precision.

Method	Recall	Precision
Our method HCL	0.65	0.19
Our method RGB	0.62	0.18
Adapt I	0.54	0.16
WBIS	0.53	0.16

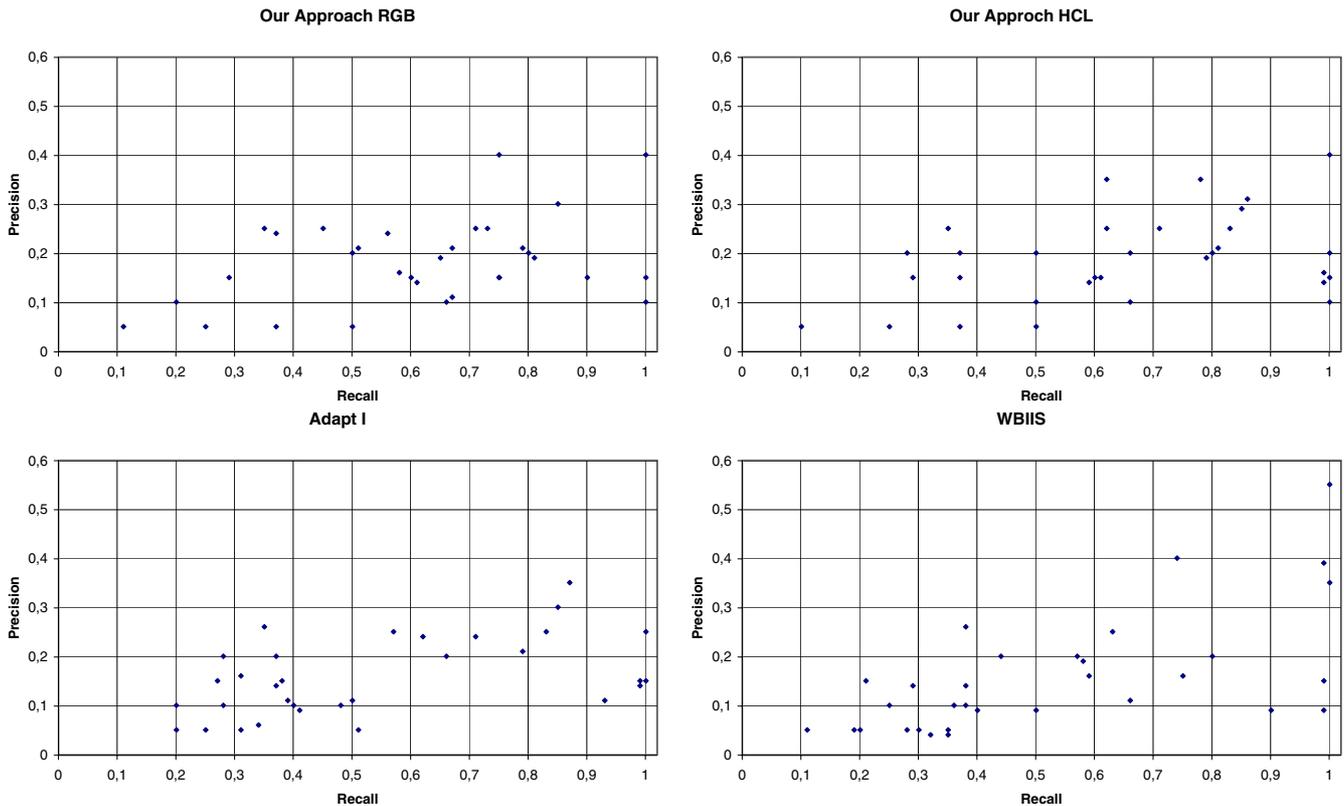


Fig. 6. Recall-precision plots of different queries for our approach and other systems. Comparing these recall-precision plots is a difficult task

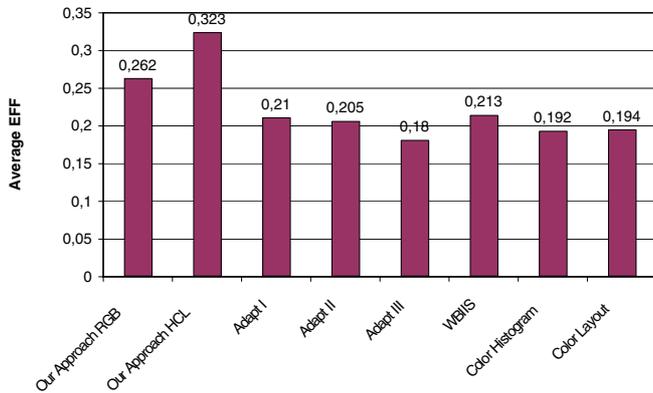


Fig. 7. Average effectiveness of several methods

The usage of the new compact effectiveness measure affirms this observation. The comparison is shown in Fig. 7. Independent of the color model our approach yields a better effectiveness, whereas using the HCL color model improves the quality even more.

6.1 Evaluation of the generalized similarity

To demonstrate the advantages of our generalized similarity, we included a number of additional images into the database. The additional images are clippings of the original images such as those shown in Fig. 2. While the similarity of the

clipped images is obvious for the human, the corresponding histograms do not show much similarity (Fig. 2b), and a similarity search based on the full histogram does not provide the desired results. We computed the similarity based on different subdivisions resp. different Haar wavelet coefficients d^k to make the comparison. The generalized similarity as defined in Sect. 4.4 was used. We compare the results with our search method used in the above comparison (d^{all} = all detail levels) and with the approach denoted by Adapt I (*AI*). Because we were not able to integrate the additional images into the online database at Stanford, the WBIS method was not tested here.

Table 1 shows the ranking of the similar images gained with the different methods. The ranking “1” marks the query image. For every set of similar images the row of the best ranking is shown in bold. While the results of the “global” approach (d^{all}) does not provide the desired images, we can always find them if the correct detail level is chosen. Note that the appropriate level of detail is not the same for all examples. The table shows that finding similar images – in the sense of clippings – is robust against the detail level. Trying d^5 , d^6 , d^7 , and other single levels and then some combinations will retrieve the expected images. We verified this strategy for a broad range of images/clippings and both color models. The results clearly demonstrate the usefulness of our flexible similarity search. Table 1 presents results yielded with the RGB color model, but the results with the HCL color model are still the same.

In addition, the flexibility of our approach allows us to focus on different aspects in the similarity search. If we use the lower coefficients in the search, the results provide a high-

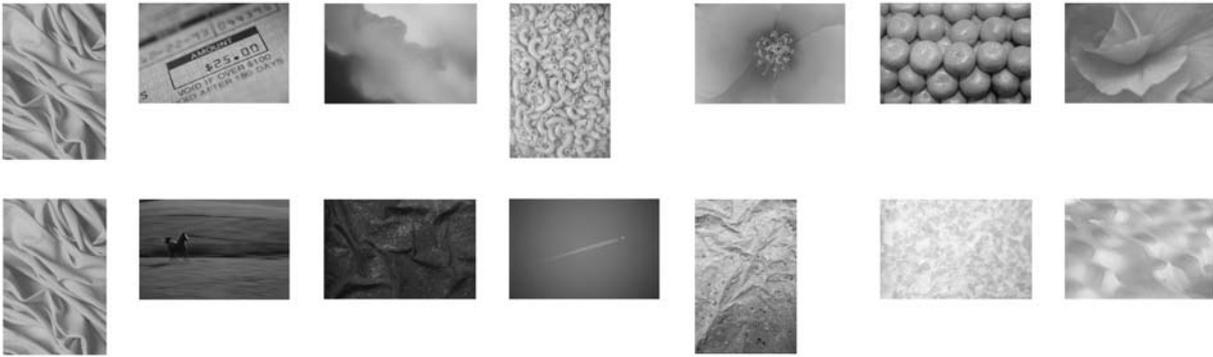


Fig. 8. The influence of choosing the level of similarity. The *left* image is the query image; the images on the *right side* are the results. The *upper row* shows similar images found when searching with d^1 ; the *lower row* shows similar images found when searching with d^5

Table 1. Rankings of the similar images (clippings) using several methods

image								
AI	1	>20	1	>20	>20	1	>20	>20
d^{all}	1	18	1	2	>100	1	>100	55
d^4	1	4	1	2	>100	1	>100	>100
d^5	1	2	1	2	4	1	>100	8
d^6	1	12	1	2	3	1	>100	2
d^7	1	19	1	2	5	1	3	2

level color similarity disregarding finer structural differences in the images. If we use the higher coefficients in the search, the results disregard the overall color but focus on the nuances of the color distribution.

In the next experiment, we used images from the Corel Photo CD database containing GIF files. The 32-bit representation of the wavelet coefficients of the RGB histograms were used. Though GIF files use a color palette of at most 256 colors and thus the histograms are rather sparse, our method works just as well with full color data. Figure 8 shows the different results obtained with detail coefficients d^1 and d^5 . Because the average information of the considered subhistograms is not contained in the higher detail coefficients, the search on those coefficients does not consider the color level of the original histogram frequencies. The coarse subdivision of the details d^1 leads to very colorlike images, while the detail coefficients of d^5 focuses on the finer nuances of the color distribution. Using the higher detail levels it is rather likely that pictures will be obtained that have a similar distribution of the subhistograms but do not share the average. As a result, we get images with different overall color hues but that seem to have a similar type of texture. The interpretation of the HCL color model is quite different. Although searching images on different levels with the HCL color model yields better results (Fig. 9), the results cannot be interpreted the same way as in Fig. 8. Therefore, the conclusions of this experiment are valid only for the RGB model.

Now we show the average effectiveness \overline{EFF} for different levels of detail and some combinations. The dependencies can be seen in Fig. 9. The results show that, for the RGB color

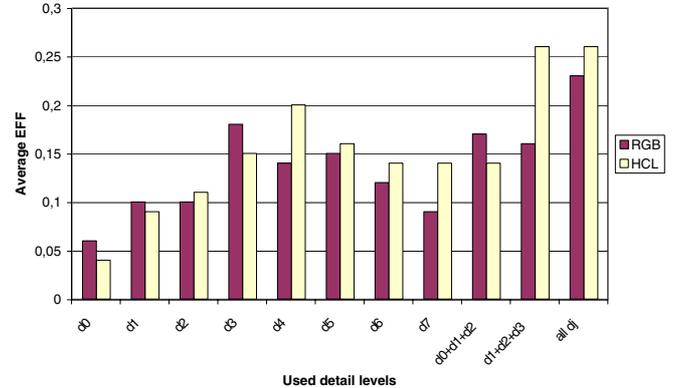


Fig. 9. Influence of the detail levels on the average effectiveness

model detail level d^3 , but for the HCL color model detail level d^4 , seems to be the most relevant. For the RGB color model one can see that adding some levels does not improve the result. Only all detail levels together perform better than d^3 does. In contrast, adding d^1 and d^2 to d^4 will improve the result when the HCL color model is used. But all levels together still perform better. Comparing the color models one can see that the HCL model performs better than the RGB model; in most cases both color models yield the same effectiveness or the HCL model clearly gives better results. This confirms our observation from Figs. 6 and 7.

Now we come back to the question of choosing an appropriate combination of levels. There is no final answer to this

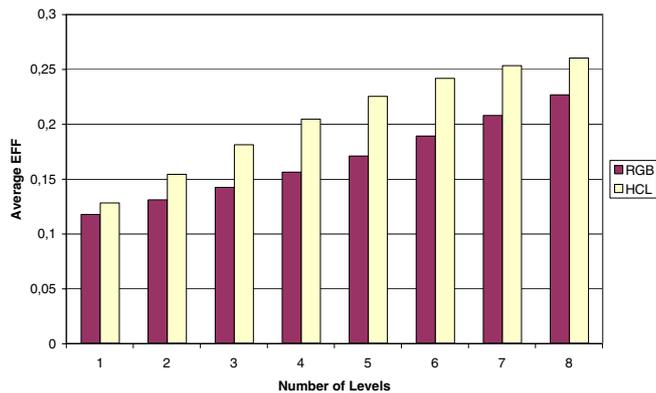


Fig. 10. Average effectiveness for different numbers of levels used

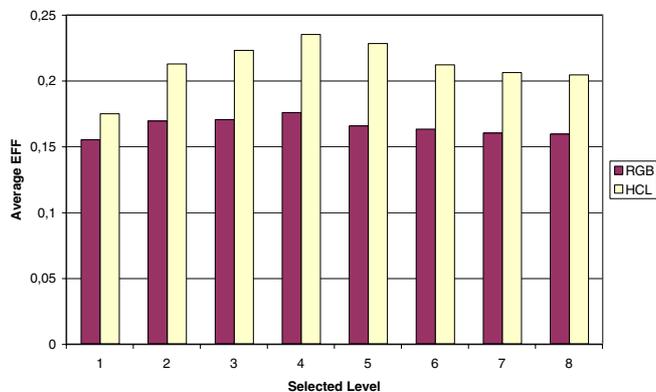


Fig. 11. Average effectiveness when the given level is included in the combination

question. We propose to start the search with all levels and then to discard some levels in order to refine the result. This approach is justified by Fig. 10. For our set of sample queries we tried out all combinations of levels and computed the average effectiveness for each number of the selected levels. One can see that the effectiveness increases with the number selected levels. Nevertheless, we found that some combinations with less than eight levels yield a higher effectiveness than all levels.

The question remains as to which levels to discard after the initial search. For each single level we compute the average effectiveness over all combinations that include this level. In Fig. 11 one can see that combinations that include level d^3 and d^4 yield the highest effectiveness, i.e., those levels are very important for an effective search. There is no final answer regarding the best combination of levels, but we have shown an efficient heuristic for how to get a good one.

Our experiments show that our approach provides a significantly better effectiveness for a conventional image similarity search. Figure 12 shows some sample queries. In addition, our generalized similarity measure supports new types of similarity queries that, to our knowledge, cannot be handled by other approaches. Example applications include the search for images that are clippings, extensions, or partially scaled versions of the desired image and the search for images with a similar texture even without a direct color similarity.

The effectiveness of a similarity measure also depends on the invariances with respect to transformations. Because color histograms do not contain any spatial information, our approach is inherently invariant toward translation and rotation. Invariance with respect to scaling can be achieved by the sub-histograms and their transformation. The influence of equally distributed noise was not evaluated, but theoretically this has no effect on the proportions (like scaling). The robustness against shifts in color, brightness, saturation, or other features can be gained by using color models separating those features (e.g., HCL) and weighting them.

6.2 Efficiency

In addition to a good effectiveness, the efficiency of an approach is important. In this subsection we therefore show that our approach provides a competitive performance. Depending on the speed requirements, our current implementation provides the choice between different performance levels. Here we use an approach based on compression similar to the one reported in [38]. The basic idea is to reduce the number of bytes used for each coefficient. This allows one to significantly improve the performance without a measurable loss of effectiveness.

In the standard mode the search uses 32 bits for each coefficient. Using a finer level of similarity or combining more levels increases the amount of required space considerably. The compressed versions of the coefficients use 1, 2, 4, or 8 bits. The values of the compressed coefficients for a given number of bits are determined as follows:

- 1 bit = $[0, 1]$: a threshold s is chosen and the bit is set if the absolute value is larger than the threshold.
- 2 bit = $[-1, 1]$: a threshold s is chosen and the first bit is set if the absolute value is larger than s and the second bit is set if the value is positive.
- 4 bit = $[-7, 7]$: a threshold s is chosen and $[-s, s]$ is mapped to $[-7, 7]$. We assign -7 or 7 for all other values with an absolute value below $-s$ resp. over s .
- 8 bit = $[-127, 127]$: analogous to 4 bit.

Now we show how the number of bits used influences the time and the effectiveness. The setting of this experiment is as follows. The number of bits per coefficient is changed for the following selections of the levels of similarity. First, we select all detail levels, and second, we select only one detail level, namely, d^3 or d^4 . Figure 13 contains the resulting effectiveness measure EFF . The experiments show that a significant compression of the detail coefficients does not compromise the effectiveness. More interesting: A run with 8-bit coefficients seems to slightly increase the effectiveness. The observations from Fig. 9 are still valid: When selecting d^3 the RGB color model is more efficient, when selecting d^4 the HCL color model is more efficient, and selecting all detail levels yields the highest effectiveness, whereas the HCL color model is the better one. In this experiment we found an appropriate threshold s as follows: For each setting of detail levels we tested several values of s and chose the one that gave the highest average effectiveness over our set of sample queries.

The bytes needed for one data entry and the time for a query for the RGB color model on our database is shown in Table 2.

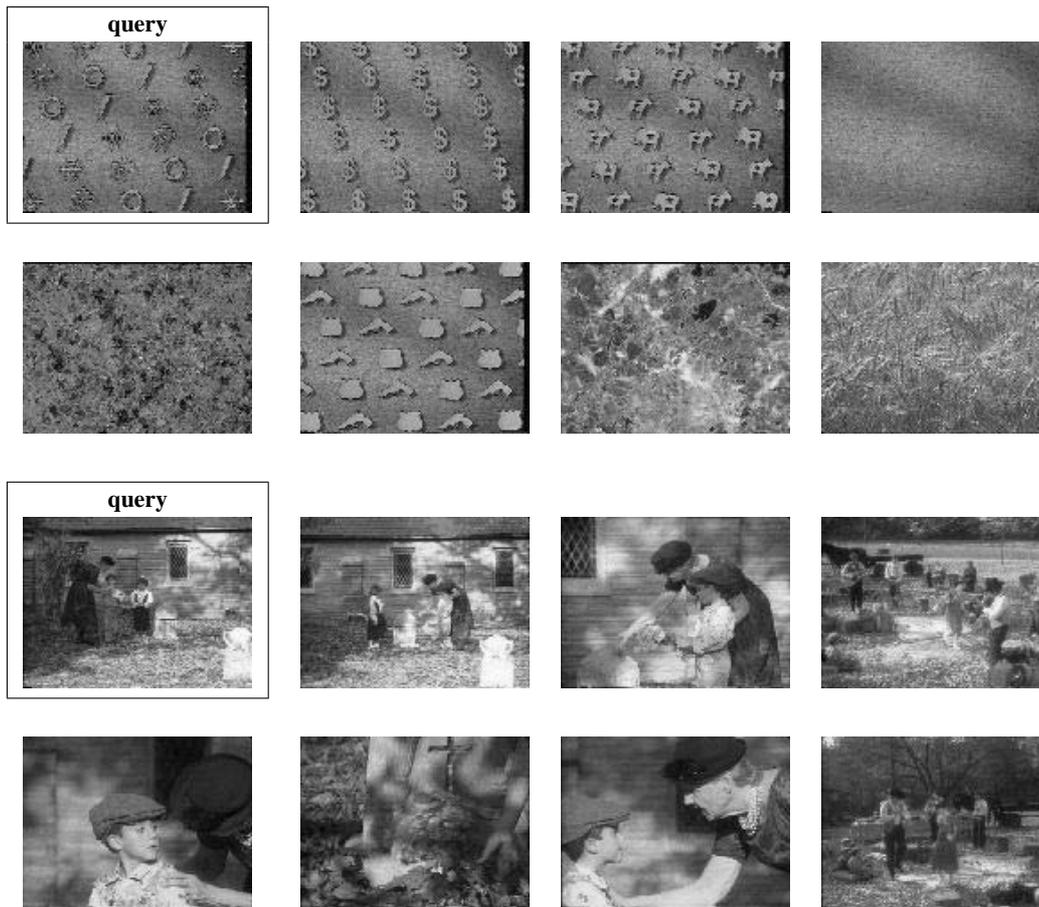


Fig. 12. Example query. The query image is the *top left* image; returned images are ordered descending row by row from *left to right*

Clearly, one can see that an increasing number of bits increases the execution time. Currently we just use a linear scan of the coefficients to determine the matching data items. For additional speedup, advanced high-dimensional index structures with sublinear performance [4] may be used.

The number of bytes needed per image for different combinations of detail levels can be calculated with

$$3 \cdot \text{\#used coefficients} \cdot \text{\#bytes per coefficient}.$$

When using all levels of detail (255 coefficients) with 32 bits (4 bytes) we need 3060 bytes per images, but using only 2 bits (0.25 bytes) requires 191.25 bytes.

As shown in the previous section, the effectiveness can be increased if multiple detail levels are combined. This, however, decreases the efficiency of the search, and therefore we have a classical effectiveness-efficiency tradeoff. Our final experiment shows this efficiency-effectiveness tradeoff. Here we compare the effectiveness of arbitrary combinations of detail levels and plot them against the number of features needed for the search, which directly corresponds to the execution time. Figure 14 shows that there is a clear tradeoff between execution time and search effectiveness.

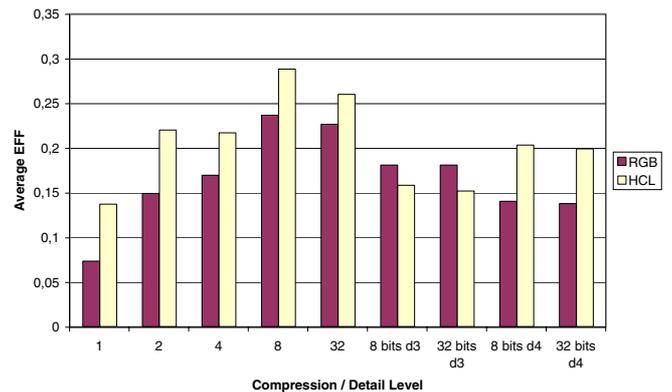


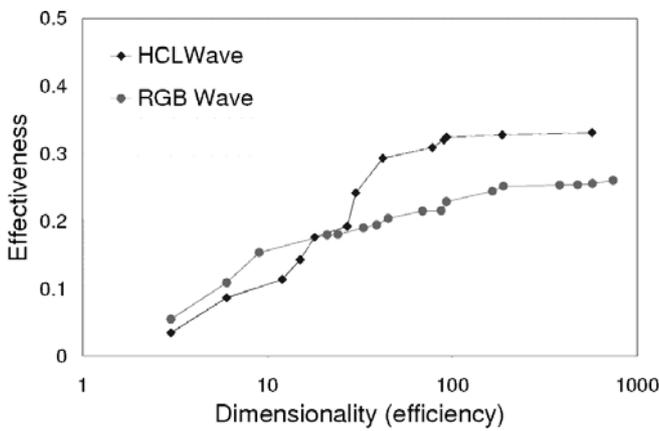
Fig. 13. Average effectiveness measure when changing the compression factor

7 Summary and conclusion

In this paper we introduce a generalized histogram-based similarity measure for an efficient and effective image similarity search. Our new approach uses a recursive subdivision of the histograms to allow a flexible search on multiple levels of color distribution details. The open question of choosing the appropriated level(s) of detail cannot be answered in general;

Table 2. Query execution time and effectiveness for different numbers of bits per coefficient

	Bytes per datum	Time
1 bit	95.63	0.19 s
2 bit	191.25	0.38 s
4 bit	382.50	0.75 s
8 bit	765.00	1.50 s
8 bit d^3 only	24.00	0.05 s
8 bit d^4 only	48.00	0.05 s
32 bit	3060.00	6.00 s
32 bit d^3 only	96.00	0.19 s
32 bit d^4 only	192.00	0.18 s

**Fig. 14.** Efficiency-effectiveness tradeoff

it depends on the user and the required kind of similarity. To choose a subdivision and the levels of detail automatically or to support the user selection is a subject of further research. Despite this, our new approach provides a significantly better effectiveness than existing systems while retaining a competitive performance.

References

- Ashley J, Flickner M, Hafner JL, Lee D, Niblack W, Petkovic D (1995) The query by image content (QBIC) system. In: Proceedings of the ACM SIGMOD conference, p 475. ACM Press, New York
- Aslandogan YA, Thier C, Yu CT, Zou J, Risse N (1997) Using semantic contents and WordNet(TM) in image retrieval. In: Proceedings of the ACM SIGIR conference. ACM Press, New York
- Berchtold S, Kriegel H-P (1997) S3: Similarity search in CAD database systems. In: Proceedings of the ACM SIGMOD conference, pp 564–567. ACM Press, New York
- Berchtold S, Böhm C, Keim DA (2001) High-dimensional indexing – improving the performance of multimedia databases. *ACM Comput Surv* 33(3):322–373
- Berretti S, Del Bimbo A, Vicario E (1999) Weighting spatial arrangement of colors in content based image retrieval. In: Proceedings of the IEEE international conference on multimedia computing and systems (ICMCS), Florence, Italy, 7–11 June 1999. IEEE Press, New York, pp 845–849
- Brunelli R, Mich O (1999) On the use of histograms for image retrieval. In: Proceedings of the IEEE international conference on multimedia computing and systems (ICMCS), Florence, Italy, 7–11 June 1999. IEEE Press, New York, pp 7–11
- Cinque L, Levialdi S, Olsen KA, Pellican A (1999) Color-based image retrieval using spatial-chromatic histograms. In: Proceedings of the IEEE international conference on multimedia computing and systems (ICMCS), Florence, Italy, 7–11 June 1999. IEEE Press, New York, pp 969–973
- Colombo C, Rizzi A, Genovesi I (1997) Histogram families for color-based retrieval in image databases. In: Proceedings of the 9th international conference on image analysis and processing (ICIAP '97), Florence, Italy, 17–19 September 1997. Lecture notes in computer science, vol 1310. Springer, Berlin Heidelberg New York, pp 204–211
- Colombo C, Del Bimbo A, Genovesi I (1998a) Interactive image retrieval by color distributions. In: Proceedings of the IEEE international conference on multimedia computing and systems (ICMCS), Austin, TX, 28 June–1 July 1998. IEEE Press, New York, pp 255–258
- Colombo C, Del Bimbo A, Genovesi I (1998b) Interactive image retrieval by color distributions. In: Proceedings of the IEEE international conference on multimedia computing and systems (ICMCS), Austin, TX, 28 June–1 July 1998. IEEE Press, New York, pp 255–258
- Corbis Corp (2001) The place for pictures online. <http://www.corbis.com>
- Flickner M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, Gorkani M, Hafner J, Lee D, Petkovic D, Steele D, Yanker P (1995) Query by image and video content: the qbic system. *IEEE Comput* 28(9):23–32
- Gevers T, Smeulders AWM (1997) Pictoseek: a content-based image search system for the world wide web. In: Proceedings of SPIE Visual '97, San Jose, CA, February 1997
- Google (2001) Google image search. <http://www.google.com/imghp?hl=en>
- Heczko M (2002) Multiresolution similarity search in image databases. <http://dbvis.inf.uni-konstanz.de/research/projects/SimSearch>
- Huang J, Kumar SR, Mitra M, Zhu W-J, Zabih, R ((1997) Image indexing using color correlograms. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 762–768
- Keim DA, Heczko M, Weber R (2000) Analysis of the effectiveness-efficiency dependence for image retrieval. In: Proceedings of the 1st DELOS Network of Excellence workshop on information seeking, searching and querying in digital libraries, Zurich, Switzerland
- Latecki L, Lakämper R (1999) Contour-based shape similarity. In: Huijsmans DP, Smeulders AWM (eds) Lecture notes in computer science, vol 1614. Springer, Berlin Heidelberg New York, pp 617–624
- Lu G, Sajjanhar A (1999) Region-based shape representation and similarity measure suitable for content-based image retrieval. *Multimedia Sys* 7(2):165–174
- Müller H, Müller W, Squire DMcG, Marchand-Maillet S, Pun T (2001) Performance evaluation in content-based image retrieval: overview and proposals. *Patt Recog Lett* 22(5):593–601
- Natsev A, Rastogi R, Shim K (1999) WALRUS: a similarity retrieval algorithm for image databases. In: Proceedings of the ACM SIGMOD conference, Philadelphia, 1–3 June 1999. ACM Press, New York, pp 395–406

22. Pass G, Zabih R, Miller J (1996) Comparing images using color coherence vectors. In: Proceedings of ACM Multimedia, Boston, 18–22 November 1996, pp 65–73
23. Pass G, Zabih R (1999) Comparing images using joint histograms. *Multimedia Sys* 7:234–240
24. Rao AR, Bhushan N, Lohse GL (1996) Relationship between texture terms and texture images: a study in human texture perception. In: Proceedings of Storage and Retrieval for Image and Video Databases (SPIE), pp 206–214
25. Seidl T (1997) Color similarity search.
<http://www.dbs.informatik.uni-muenchen.de/cgi-bin/similarity/color/HistoWWW>, <http://www.dbs.informatik.uni-muenchen.de/cgi-bin/similarity/color/ctest>
26. Seidl T, Kriegel H-P (1997) Efficient user-adaptable similarity search in large multimedia databases. In: Proceedings of the international conference on very large databases, Athens, Greece, 26–29 August 1997, pp 506–515
27. Stehling RO, Nascimento MA, Falco AX (2002) A compact and efficient image retrieval approach based on border/interior pixel classification. In: Proceedings of the 11th ACM international conference on information and knowledge management (CIKM), McLean, VA, 4–9 November 2002, pp 102–109
28. Stollnitz EJ, DeRose TD, Salesin DH (1996) Wavelets for computer graphics, theory and applications. Morgan Kaufmann, San Francisco
29. Stricker M, Swain M (1994) The capacity of color histogram indexing. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Seattle, June 1994, pp 704–708
30. White DA, Jain RC (1997) ImageGREP: Fast visual pattern matching in image databases. In: Proceedings of Storage and Retrieval for Image and Video Databases (SPIE), pp 96–107
31. Wan X, Kuo C-CJ (1996) Color distribution analysis and quantization for image retrieval. In: Proceedings of Storage and Retrieval for Image and Video Databases (SPIE), pp 8–16
32. Wang JZ (2002a) Content-based image retrieval project. <http://www-db.stanford.edu/IMAGE/>
33. Wang JZ (2002b) Image database.
<http://wang.ist.psu.edu/docs/related/>
34. Wang JZ, Wiederhold G, Firschein O (1997a) System for screening objectionable images using daubechies' wavelets and color histograms. In: Steinmetz R, Wolf LC (eds) Lecture notes in computer science, vol 1309. Springer, Berlin Heidelberg New York, pp 20–30
35. Wang JZ, Wiederhold G, Firschein O, Wei SX (1997b) Content-based image indexing and searching using daubechies' wavelets. *Int J Digital Libr* 1(4):311–328
36. Wang JZ, Wiederhold G, Firschein O, Wei SX (1997c) Wavelet-based image indexing techniques with partial sketch retrieval capability. In: Proceedings of the 4th forum on research and technology advances in digital libraries (ADL'97), Washington, DC, pp 13–24
37. Wang JZ, Wiederhold G, Li J (1998) Wavelet-based progressive transmission and security filtering for medical image distribution. In: Wong S (ed) Medical image databases. International series in engineering and computer science, sects 465. Kluwer, Dordrecht, pp 303–324
38. Weber R, Schek H-J, Blott S (1998) A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: Proceedings of the 24th international conference on very large databases, New York, 24–27 August 1998
39. Weber R, Boehm K, Schek H-J (2000) Interactive-time similarity search for large image collections using parallel va-file. In: Proceedings of the international conference on data engineering (ICDE 2000), San Diego, pp 197–197
40. You J, Shen H, Cohen HA (1997) An efficient parallel texture classification for image retrieval. *J Vis Lang Comput* 8(3):259–372
41. Zhang A, Cheng B, Acharya R (1995) Texture-based image retrieval in image database systems. In: Revell N, Tjoa AM (eds) In: Proceedings of the 6th international conference on database and expert systems applications (DEXA'95), London, 4–8 September 1995. ONMIPRESS, San Mateo, CA, pp 349–356