# Cortina: A System for Large-scale, Content-based Web Image Retrieval

Till Quack, Ullrich Mönich, Lars Thiele, B.S. Manjunath
Electrical and Computer Engineering Department
University of California
Santa Barbara, CA 93106-9560
{tquack, moenich, thiele, manj}@ece.ucsb.edu

## ABSTRACT

Recent advances in processing and networking capabilities of computers have led to an accumulation of immense amounts of multimedia data such as images. One of the largest repositories for such data is the World Wide Web (WWW). We present Cortina, a large-scale image retrieval system for the WWW. It handles over 3 million images to date. The system retrieves images based on visual features and collateral text. We show that a search process which consists of an initial query-by-keyword or query-by-image and followed by relevance feedback on the visual appearance of the results is possible for large-scale data sets. We also show that it is superior to the pure text retrieval commonly used in large-scale systems. Semantic relationships in the data are explored and exploited by data mining, and multiple feature spaces are included in the search process.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.5 [**Information Storage and Retrieval**]: Online Information Services

## General Terms

Design, Documentation, Performance, Management

## Keywords

web image retrieval, online, WWW, large-scale, MPEG-7, relevance feedback, clustering, semantics, association rules

## 1. INTRODUCTION

In recent years advances in processing and networking capabilities of computers have led to an accumulation of immense amounts of data. Many of these data are available as multimedia documents, i.e. documents consisting of different types of data such as text and images. By far the largest repository for such documents is the Word Wide Web (WWW). Clearly, the amount of information available on the WWW creates enormous possibilities and challenges at the same time. Much progress has been made in both text-based search on the WWW and content-based image retrieval for research applications. However, little work has been done to combine these fields to come up with large-scale, content-based image retrieval for the WWW.

Commercial search engines like Google have successfully implemented methods [7] to improve text-based retrieval. While text-based search on the WWW has reached astounding levels of scale, commercially available search for *images* or other multimedia documents also relies on the use of text only. In addition, to our knowledge no commercial search engine offers the possibility to refine the search using relevance feedback on visual appearance, i.e. the option to look for results visually similar to an image selected from a first set of results. Content-based image retrieval (CBIR) systems introduced by the computer vision community on the other hand, usually exist in a research form: In general, the database of the proposed systems is relatively small, and the scalability of the systems has not been tested with a large, real-world dataset. Few scale to some extent, but usually only if they are restricted and tuned to a certain class of images — e.g. aerial pictures or medical images. In addition, many of these CBIR systems rely only on low level features and do not incorporate other data to improve the semantic quality of the search.

We present a system which demonstrates (to our knowledge for the first time) the scalability of selected image-retrieval methods to database sizes in the order of millions. Relevance-feedback is applied to this large dataset and the semantics within the system are explored and exploited using data mining methods. With this project we take content-based web-image retrieval to larger scales and thus one step closer to commercial applications.

The remainder of this paper is organized as follows: Section 2 introduces the system design, in section 3 the specifics of the content-based retrieval are discussed and section 4 gives a short overview over the data mining. Sections 5 and 6 contain results and conclusions, respectively.

## 2. SYSTEM OVERVIEW

This section gives an overview of our system as shown in figure 1. A web-crawler browses the WWW to collect images. The web-sites listed in the categories *Shopping* and *Recreation* of the DMOZ directory [1] are chosen as a starting point. From these web-sites we collected over 3 million JPEG images and associated text. The associated text con-


*MM'04,* October 10-16, 2004, New York, New York, USA

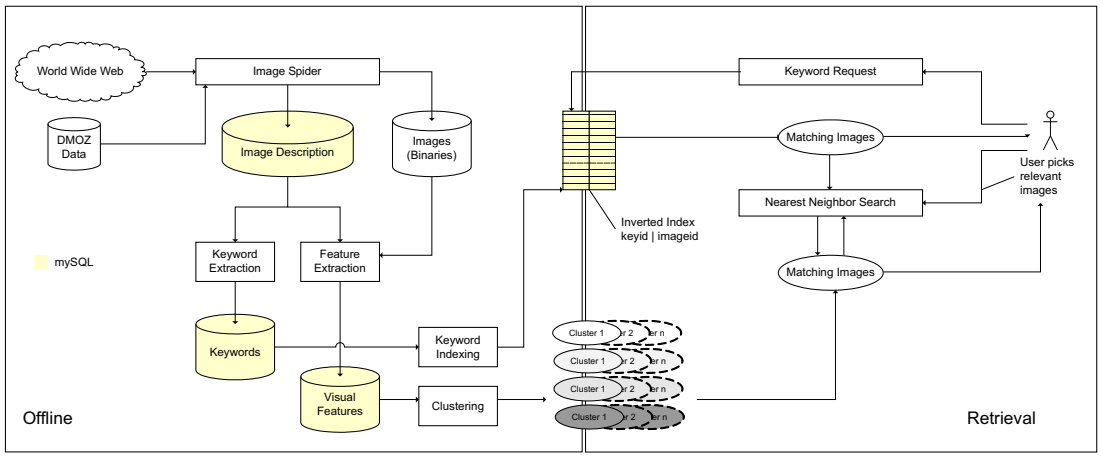


**Figure 1: System Architecture**

sists of the ALT portion of the HTML image-tag and some 50 words extracted from the text around the image. (Note that this method usually produces quite noisy results, i.e. the relation between keywords and the actual objects depicted in the image is not guaranteed). This information, along with some additional data like the image size, is stored in a relational database. The images are stored to a storage server.

For each image, the collected keywords are stemmed using the Porter Stemmer [8] and inserted into an inverted index, which maps keywords to the images and allows search by ranked boolean queries.

As visual features, four generic visual MPEG-7 descriptors are used. Since the set of images on the WWW is extremely diverse and often of lower quality and size than commonly used in image processing research, we chose four *global* feature descriptors for color and texture. Namely, the Homogeneous Texture Descriptor (HTD), the Edge Histogram Descriptor (EHD), the Scalable Color Descriptor (SCD) and the Dominant Color Descriptor (DCD) were chosen from the MPEG-7 standard descriptors [6]. The two texture descriptors were chosen because they complement each other: The EHD performs best on large non-homogeneous regions, while the HTD operates on homogeneous texture regions. The features are extracted and stored in a relational database. The whole dataset is then organized in clusters of each descriptor type in the file-system in order to reduce the search time.

While the preceding steps are performed off-line, the following on-line search-concept was implemented[1] for the retrieval process: A user enters a keyword-query through a web-interface. Matching images are retrieved with ranked boolean queries from the inverted keyword-index and returned to the client. One or more steps of relevance feedback follow: The user has the possibility to choose one or several images he thinks match best what he was looking for. From the image(s) the user chose, query vectors are constructed to perform a nearest neighbor search in the feature spaces of the visual feature descriptors. Note that this search-concept starts with the high-level semantics of a text-based search, followed by a content-based search which is based on the low-level content.

[1]http://www.cortina.ch

# 3. CONTENT-BASED RETRIEVAL

This section discusses the details of the content-based retrieval part in our system. EHD, HTD and SCD are vectors in a $n$-dimensional vector space $\mathbb{R}^n$. We used the Minkowski metrics $L_1$ and $L_2$ to measure similarity in these feature spaces. The DCD which does not represent a single vector in a vector-space needs special treatment as described in [6], which includes a custom dissimilarity measure. Retrieval in the visual feature spaces consists of K-Nearest Neighbor (K-NN) search. Since we use several visual feature-types, we need to combine the results of the retrievals of each feature-type for a joint search. We decided to combine the distances of the four descriptors in a linear way.

$$d = \alpha_{\mathrm{EHD}} d_{\mathrm{EHD}} + \alpha_{\mathrm{HTD}} d_{\mathrm{HTD}} + \alpha_{\mathrm{SCD}} d_{\mathrm{SCD}} + \alpha_{\mathrm{DCD}} d_{\mathrm{DCD}} \quad (1)$$

with $\alpha_{\mathrm{EHD}} + \alpha_{\mathrm{HTD}} + \alpha_{\mathrm{SCD}} + \alpha_{\mathrm{DCD}} = 1$ and $\alpha_f \in \mathbb{R}^+$, $f \in \{\mathrm{EHD,HTD,SCD,DCD}\}$. As the distances have different distributions ranges for each descriptor type they were normalized. We match the distribution of each feature type

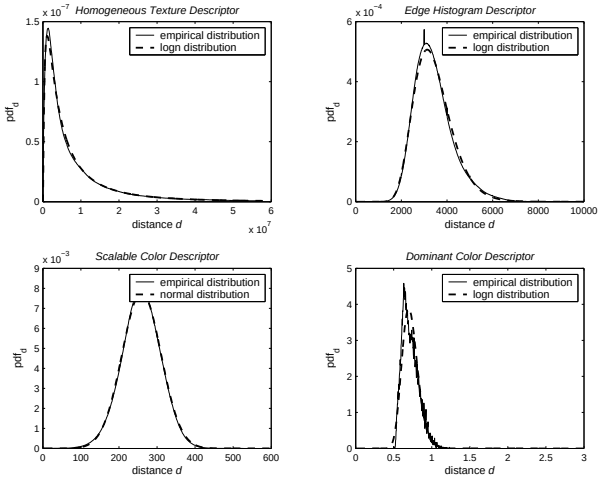


**Figure 2: Distance Distributions**

with a known distribution (normal and log-normal distributions), see figure 2. The parameters for each are determined with a Maximum-Likelihood parameter estimation based on

the whole dataset. Each of the distributions is transformed to a Gaussian distribution in the range $[0, 1]$ to be able to compare and combine the distances. The ratios $\alpha_f$ in equation (1) should be set query-dependent, i.e. for many queries, there is a descriptor type which is best suited to describe the query. Usually a user enters the system with query-by-keyword and then refines the search with relevance feedback. The weights are changed in the relevance feedback process by the information obtained from images selected by the user as relevant. The initial weights are set based on data mining, see section 4. The descriptor type which obtains the highest weight is defined as the *Primary Descriptor*, the other ones we call *Secondary Descriptors*.

To avoid a slow linear search over the whole database we cluster the feature vector data. Clustering and indexing for dimensions up to 20 is very well solved, but in high-dimensional spaces there still exists no efficient solution [3]. Note that the descriptors we use are in fact high-dimensional and that our dataset is extremely large and diverse. We use the k-means algorithm [4] to form the clusters. This choice was mainly motivated by the comparably fast processing of the k-means algorithm compared to other unsupervised clustering methods. However, calculating the cluster-centroids from the whole dataset would still use more time and resources than were available. Thus, 10% of the feature vectors are randomly sampled from the database to calculate 400 cluster centroids $c_i$ for each descriptor type. Based on experiments the number of 400 cluster-centroids offers a good trade-off between speed and accuracy. Using a joint distance as defined in equation (1) complicates the problem of retrieval in the clustered dataset: Each feature-type is clustered individually, but for a given query, there might be only a very small or no common subset of images in the clusters that are searched for each feature type. Our solution is to search *only* the images in the cluster defined by the *Primary Descriptor* for a given query (i.e. the descriptor with the highest weight) with *all* the feature types. That means that for each cluster of each feature type we need to store also the secondary descriptors in the same order. E.g. for each EHD cluster there are three other repositories which contain the vectors of the secondary descriptors (HTD, SCD and DCD) in the same order.

A relevance feedback mechanism which scales to millions of items is implemented in Cortina. Relevance feedback starts (after a query-by-keyword) by selecting the images that are the best matches visually. Several additional iterations can follow to refine the query parameters. In other words, based on the user feedback the system decides automatically which combination of the low level features should be used, i.e. the values for $\alpha_f$ and it adapts the query vector based on the user input. In every iteration a new query representation and new set of weights are computed by using the images the user has selected as relevant, thus the system is using query vector and weight movement in real time [5]. For each feature type $f$, the new vector $\mathbf{r}_f$ in iteration $n$ is a linear descriptor combination of images marked as relevant.

$$\mathbf{r}_f = \frac{\sum_{i=1}^{M} \mathbf{r}_{f,i}}{M}$$

where $i = 1 \dots M$ indexes the relevant images. The new query vector $\mathbf{q}_f^{(n)}$ is a linear combination of the $\mathbf{r}_f$ and the old query vector $\mathbf{q}_f^{(n-1)}$:

$$\mathbf{q}_f^{(n)} = weightQuery_f \cdot \mathbf{q}_f^{(n-1)} + weightObject_f \cdot \mathbf{r}_f$$

with $weightQuery_f + weightObject_f = 1$. This updating scheme can be used for descriptors, having a vector-like layout, i.e. HTD, EHD and SCD. For the DCD we developed a different approach, which turned out to be similar to [9]. In addition to the weights $\alpha_f$ from equation (1) there are the weights $weightQuery_f$ and $weightObject_f$ to be set in each iteration. We compute the average distance $\bar{d}_f$ consisting of the not normalized distances between the query from last iteration and the selected images for feature type $f$:

$$\bar{d}_f = \frac{\sum_{i=1}^{M} \sum_{k=1}^{M} d_{f,ik}}{M^2}$$

where $i, k$ index the relevant images and $d_{f,ik}$ is the distance between image $i$ and $k$ with the distance measure defined for feature type $f$. The $\bar{d}_f$ are added and normalized to the range $[0, 1]$, such that close (small $\bar{d}_f$) images for a descriptor-type $f$ result in a higher fraction in the combined final distance. The weights are updated in iteration $n$ as follows:

$$\alpha_f^{(n)} = \xi \cdot (1 - \bar{d}_f) + \zeta \cdot \alpha_f^{(n-1)}$$

$$weightQuery_f = (1 - \bar{d}_f) \cdot \beta$$

We set $\beta$ to 0.25, $\xi$ to 0.8 and $\zeta$ to 0.2 based on experiments.

# 4. DATA MINING FOR SEMANTICS

To improve the retrieval-results based on semantic associations between text and visual features, association rule mining is applied [2]. Transactions were formulated as a conjunction of keywords and visual features, i.e. each image is considered to be a transaction consisting of a series of keywords and cluster identifiers. Note that by representing the visual information by the cluster-id instead of feature vectors, we achieve a dimensionality reduction which enables the mining for semantic associations. In fact, we believe this form of association rule mining is one of the few, if not the only method which enables the discovery of semantic relations in a dataset of this scale, since well-known state-of-the-art algorithms with linear scalability are available.

The insights gained by mining these associations are used to select clusters for k-NN search. Given, that a particular keyword was used to query the database initially, for the following visual k-NN searches we choose those visual feature clusters which lie both close to the query vector and have a high degree of association with the keyword. That means that we are able to select clusters not only based on geometric measures, which are difficult to apply to high dimensional spaces. This results in the following steps:

1. Look up the association rules for the initial query-keyword.
2. Calculate the distances from the query-image to the centroids of the set of clusters $\mathcal{C}_A$ given by the association rules.
3. Load the $n$ closest clusters from $\mathcal{C}_A$ and the cluster closest to the query vector if it is not already given by the association rules. (We usually set $n$ to 3).
4. Do a similarity search on these $n$ clusters.

Another benefit of mining rules which associate keywords with visual features is that we can set initial weights for the combination of the different visual feature types, given that a particular keyword was used to query the database initially. In other words, if we find that the associations between a particular keyword and the clusters of a particular feature type are stronger than the associations between the keyword and the other feature types, we set the weights in equation (1) accordingly. Early results suggest that using frequent itemset mining and association rules to explore and exploit the semantics as described above is a technique very suitable for large-scale datasets like Cortina, and it results in improvements in the retrieval precision.

## 5. RESULTS

A fully working system has been designed, implemented and is available to the public [2]. In unsupervised, large-scale systems it is difficult to measure the quality. There is no ground truth and the common precision/recall measures can not be applied. We thus define a precision measure which is based on user surveys. Users are given a set of keywords and asked to query the database. From the initial set of images they have to select the image(s) they think are closest to a specific semantic concept of their choice (e.g. for the keyword *shoe* a black leather shoe) and execute two steps of relevance feedback. The precision P is measured with

$$P = \frac{|\mathcal{C}|}{|\mathcal{A}| \in \{8, 16\}} \tag{2}$$

i.e. the correct results $\mathcal{C}$ in the first 8 or 16 ($\mathcal{A}$) returned images are to be counted. The term *correct* is defined as *visually very similar to the specific concept the user selected*. $P$ is shown in figure 3 for the averaged results for the keywords {shoe, bike, mountain, porsche, bikini, flower, donald duck, bird, sunglass, ferrari, digital camera} – $P$ along the y-axis, along the x-axis are the steps of the experimental queries. It can be seen that visual nearest neighbor search and relevance feedback improve the precision compared to pure keyword query results if the user is looking for a specific semantic concept. Especially the improvements from the initial results to the first round of relevance feedback are significant.

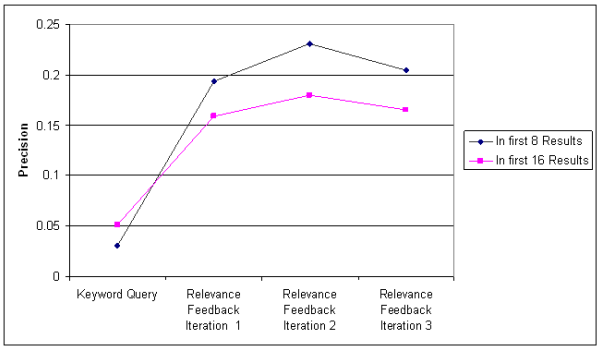


**Figure 3: Precision from user questioning**

[2]http://www.cortina.ch

## 6. CONCLUSIONS

A system for large-scale, content-based image retrieval on the WWW is introduced and implemented. With over 3 Million images this is the first CBIR system that we are aware of which scales to more than one Million images. To achieve this large-scale experiment, a dataset consisting of images and collateral text was collected from the WWW. Several low-level MPEG-7 visual feature types are jointly used for content-based retrieval. Keywords are indexed as an additional high-level feature.

The retrieval-concept of a keyword-query followed by relevance feedback on visual appearance was shown to result in more precise results than plain keyword search. Association rule mining is applied to the domain of large-scale multimedia data find semantic relationships between keywords and visual features and is exploited to improve the search results.

## 7. ACKNOWLEDGMENTS

This project is supported in part by grants from the US Office of Naval Research ONR #N00014-01-1-0391, ONR #N00014-04-1-0121, and an infrastructure grant from the NSF #EIA-0080134. Till Quack's visit to UCSB was supported in part by a fellowship from ETH Zurich, Switzerland.

## 8. REFERENCES

[1] Dmoz open directory project, http://www.dmoz.org.

[2] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

[3] E. Chávez and G. Navarro. A probabilistic spell for the curse of dimensionality. In *Revised Papers from the Third International Workshop on Algorithm Engineering and Experimentation*, pages 147–160. Springer-Verlag, 2001.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., 2nd edition, 2000.

[5] D. Heesch and S. Rüger. Performance boosting with three mouse clicks - relevance feedback for cbir. In *25th European Conference on Information Retrieval Research (ECIR, Pisa, Italy, 14-16 Apr 2003)*, pages pp 363–376, 2003.

[6] B. S. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG7: Multimedia Content Description Language*. 1st edition, 2002.

[7] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[8] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[9] K.-M. Wong and L.-M. Po. MPEG-7 dominant color descriptor based relevance feedback using merged palette histogram. In *IEEE International Conference on Speech, Acoustics, and Signal Processing*, 2004.