Content-Based Image Retrieval Systems: A Survey

Remco C. Veltkamp, Mirela Tanase
Department of Computing Science, Utrecht University
email: {Remco.Veltkamp,mirela}@cs.uu.nl

October 2000

1 Introduction

In many areas of commerce, government, academia, and hospitals, large collections of digital images are being created. Many of these collections are the product of digitizing existing collections of analogue photographs, diagrams, drawings, paintings, and prints. Usually, the only way of searching these collections was by keyword indexing, or simply by browsing. Digital images databases however, open the way to content-based searching. In this paper we survey some technical aspects of current content-based image retrieval systems.

A number of other overviews on image database systems, image retrieval, or multimedia information systems have been published, see e.g. [TY84], [Gro94], [GR95], [Jai96], [EG99], [RHC99]. This survey however, is about the functionality of temporary image retrieval systems in terms of technical aspects: querying, relevance feedback, features, matching, indexing data structures, and result presentation.

A number of keyword-based general WWW search engines allows to indicate that the media type must be images, see for example HotBot (`http://hotbot.lycos.com/`), and NBCi (`http://www.nci.com/`). A number of other general search engines are more specifically for images, such as Yahoo!'s Image Surfer (`http://isurf.yahoo.com/`) or the multimedia searcher of Lycos (`http://multimedia.lycos.com/`), but they are still only keyword based. There are many special image collections on the web that can be searched with a number of alphanumerical keys. For example, ImageFinder (`http://sunsite.berkeley.edu/ImageFinder/`) provides a list of such collections as a tool to help teachers locate historical photographs from collections around the world.

AltaVista Photofinder (see below) is a search engine that allows content-based image retrieval, both from special collections, and from the Web. In the remainder of this paper, we will give an overview of such content-based image retrieval systems, both commercial/production systems, and research/demonstration systems.

2 Criteria

Many image retrieval systems can be conceptually described by the framework depicted in figure 1. In this article we survey how the user can formulate a query, whether and how relevance feedback is possible, what kind of features are used, how features from query image and data base image are matched, what indexing data structures are used, and how the retrieval results are presented to the user.

The user interface typically consists of a query formulation part and a result presentation part. Specification of which images to retrieve from the database can be done in many ways. One is to browse through the database one by one. Another way is to specify the image in terms of keywords, or in terms of image features that are extracted from the image, such as a color histogram. Yet another way is to provide an image or sketch from which features of the same type must be extracted as for the database images, in order to match these features. A nice taxonomy of

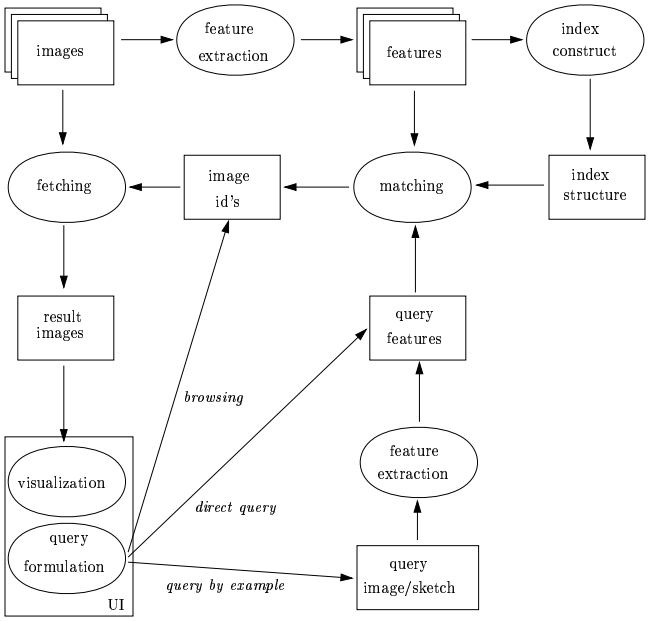

Figure 1: Content-based image retrieval framework.

interaction models is given in [Ven97]. Relevance feedback is about providing positive or negative feedback about the retrieval result, so that the system can refine the search.

We will consider several classes of features that are used to specify queries: color, textures, shape, spatial layout, and faces. Color features are often easily obtained directly from the pixel intensities, e.g. color histogram over the whole image, over a fixed subimage, or over a segmented region are often used. Although a precise definition of texture is untraceable, the notion of texture generally refers to the presence of a spatial pattern that has some properties of homogeneity. In particular, the homogeneity cannot result from the presence of only a single color in the regions, but requires interaction of various colors.

There is no universal definition of what shape is either. Impressions of shape can be conveyed by color or intensity patterns, or texture, from which a geometrical representation can be derived. This is shown already in Plato's work Meno [PlaBC]. (This is one of the so-called Socratic dialogues, where two persons discuss aspects of virtue; to memorialize Socrates, one of the figures is called after him.) In this work, the word 'figure' is used for shape. Socrates' description

> "figure is the only existing thing that is found always following color"

does not satisfy Meno, after which Socrates gives a definition in "terms employed in geometrical problems":

> "figure is limit of solid".

In this paper too we consider shape as something geometrical. Therefore, we consider edge orientation over all pixels as texture, but edge orientation at only region contours as shape information. Shape descriptors are diverse, e.g. turning angle functions, deformable templates, algebraic moments, and Fourier coefficients. For an overview of shape matching techniques, see [VH99].

Spatial layout is about the absolute or relative position of color, texture, or shape information. Higher level features are increasingly more specific, and thus less widely used. However, faces are frequently present in pictures and relatively often used as a feature, so that we tally its use separately.

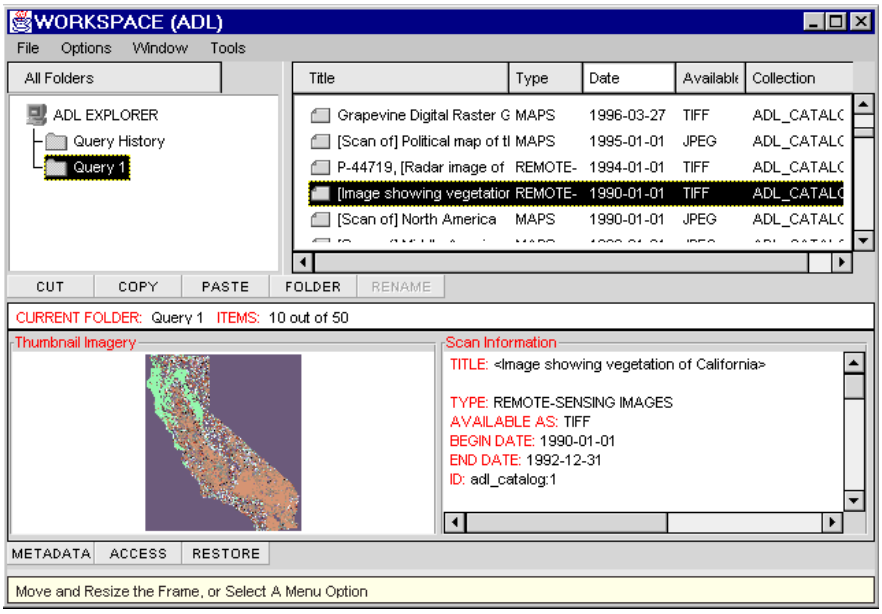

Figure 2: ADL workspace window, showing the query and the resulting thumbnails.

3 Systems

Below we describe a number of content-based image retrieval systems, in alphabetical order. If no *querying*, *features*, *matching*, *indexing* data structure, or *result presentation* is mentioned, than it is not applicable to the system (e.g. there is no relevance feedback), or no such information is known to us (e.g. often no information is given about indexing data structures).

1 ADL (Alexandria Digital Library)

Developer University of California, Santa Barbara.

URL The homepage is at `http://www.alexandria.ucsb.edu/adl.html`. A walkthrough demo is available at `http://alexandria.ucsb.edu/adljigi/tutorials/walkthrough1/`.

References [Man95], [WHS99].

Querying With a map browser the user can interactively pan and zoom a two-dimensional map of the world to locate his areas of interest, and select a query area that must contain or overlap with the database images. Then, the user can select a catalogue of images, set alphanumeric query parameters such as type (e.g. aerial photos, maps, remote sensing images), and then retrieve the images overlapping with the area indicated with the map browser.

Features Although it has been reported [MM99] that images can be searched with texture features, the ADL website does not show this capability. What remains is searching on keywords.

Matching When the query is sent to the databases, the query area is compared to the object footprints, according to the functioning matching operator ('contains' or 'overlaps'), and those objects that match are potentially members of the result set for the query.

Result presentation The matched images are shown as thumbnails, see figure 2, and their footprints are shown on the map in the map browser.

Applications An important focus for ADL's collection is on information supporting basic science, including the Earth and Social Sciences. The image datasets (will) include digital elevation models (DEMs), digital raster graphics (DRGs), scanned aerial photographs, Landsat images, seismic datasets, Sierra Nevada Ecologic Project datasets, and Mojave Ecologic Project datasets.

2 AltaVista Photofinder

Developer The AltaVista search engine was originally developed at DEC Research Lab, and is now run by AltaVista Company.

URL http://image.altavista.com/cgi-bin/avncgi.

Features Similarity is based on visual characteristics such as dominant colors, shapes and textures. No details are given about the exact features.

Querying The user first types keywords to search for images tagged with these words. If a retrieved image is shown with a link "similar", the link gives images that are visually similar to the selected image. Similarity is based on visual characteristics such as dominant colors, shapes and textures. The user cannot set the relative weights of these features, but judging from the results, color is the predominant feature.

Indexing The system is built on the VIR Image Engine, see below.

Result presentation The retrieved images are shown as thumbnails, without an explicit order.

3 Amore (Advanced Multimedia Oriented Retrieval Engine)

Developer C & C Research Laboratories NEC USA, Inc.

URL http://www.ccrl.com/amore/.

References [MHH97], [MHH99].

Features The image is segmented into at most eight regions of homogeneous color, and downsized to 24×24 pixels. The regions in this picture are directly used for matching [HHSH93].

Querying The user first selects a category of images. An initial set of images can be selected at random or by keyword. Of these images, visually similar images can be retrieved. The query image can also be specified by its URL. In a research version of the system, sketching a query image was possible. The user can indicate the relative importance of color and shape.

Matching First a correspondence between regions in the query and target image is found. Regions corresponding to the same regions in the other image are merged. The shape similarity between two regions is based on the number of pixels of overlap. The color similarity between two regions is the distance in HLS space between the uniform region colors [HHSH93].

Indexing Indexing is performed by the COIR (Content-Oriented Image Retrieval) system, see [HMO+97].

Result presentation The retrieved images are shown as thumbnails, without an explicit order, see figure 3. In a research version of the system, result images were displayed as a scatter plot, with shape and color similarity values at the axes, or on a perspective wall [MHH97].

Applications Amore is used in the art retrieval system Arthur (Art Media and Text Hub and Retrieval System, http://www.isi.edu/cct/arthur/), developed at the Center for Cultural Technology within the Information Sciences Institute of the University of Southern California.

4 Berkeley Digital Library Project

Developer University of California, Berkeley.

URL The homepage of project is at http://elib.cs.berkeley.edu/, a demo of retrieval from all photos in the collection is at http://elib.cs.berkeley.edu/photos/all.shtml.

References [CO96].

Features There is a number of alphanumerical keys available for querying: the collection, key words, location, county, and photographer. The colors of each image are quantized into 13 colors bins. Six values are associated with each color bin: the percentage of the image with colors in that bin, and the number of 'very small', 'small', 'medium', 'large', and 'very large' dots of that color found.

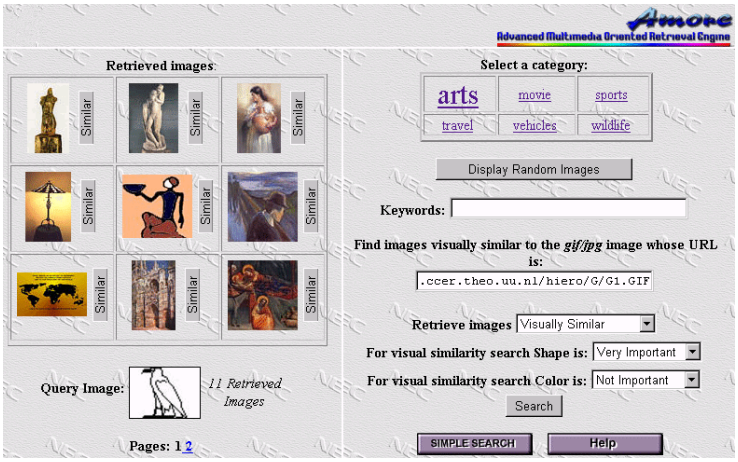

Figure 3: Amore result of similarity retrieval on shape.

Querying For content-based search, the user can select 13 colors, and indicate the amount ('any', 'partly', 'mostly') of that color in the picture. Also, for colored regions the user can indicate the size ('any', 'small', 'medium', 'large') and the quantity of regions with that color ('any', 'few', 'some', 'many').

Matching Image features are stored as text strings. For example, a picture of a sky with clouds might have a few large white regions, and a large amount of blue, and would have a feature text string "`mostly_blue large_white_few`". Matching is done by substring matching, for example with a query string "`large_white%`".

Indexing All features are put into a relational database (the Informix Universal Server database management system).

Result presentation The retrieved photos are presented unordered, with id-number, photographer, and collection keys.

Applications The collections consist of 23195 images of plants, animals, people, and landscapes, 17000 images from the California Department of Water Resources, Corel Stock photos, and aerial photos of the Sacramento river delta region.

5 Blobworld

Developer Computer Science Division, University of California, Berkeley.

URL `http://elib.cs.berkeley.edu/photos/blobworld/`. A demo of Blobworld is available at `http://elib.cs.berkeley.edu/photos/blobworld/start.html`.

References [CTB+99].

Features The features used for querying are the color, texture, location, and shape of regions (blobs) and of the background. The color is described by a histogram of 218 bins of the color coordinates in Lab-space. Texture is represented by mean contrast and anisotropy over the region. Shape is represented by (approximate) area, eccentricity, and orientation.

Querying The user first selects a category, which already limits the search space. In an initial image, the user selects a region (blob), and indicates the importance of the blob ('somewhat', 'very'). Next the user indicates the importance of the blob's color, texture, location, and shape ('not', 'somewhat', 'very'). More than one regions can be used for querying.

Matching To match two color histograms h_1 and h_2, the quadratic form distance is used: $d(h_1, h_2) = (h_1 - h_2)^T A (h_1 - h_2)$, where $A = (a_{ij})$ is a symmetric matrix of weights representing the similarity between color bins i and j. The distance between two texture descriptors is the Euclidean distance between their values of $(contrast, contrast \times anisotropy)$. The distance between centroids is the Euclidean distance. The distances are combined into a single final distance.

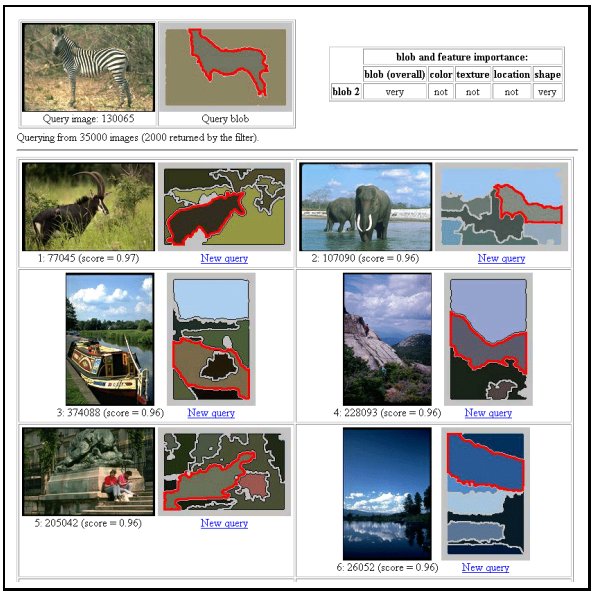

Figure 4: Blobworld.

Indexing Rather than actually computing the distances between the full color histogram vectors of length 218 as $d(h_1, h_2) = (h_1 - h_2)^T A(h_1 - h_2)$, singular value decomposition (SVD) is used to project the histogram vectors onto a lower-dimensional subspace. The resulting points are indexed by an R*-tree [BKSS90].

Result presentation The retrieved images are ranked in linear order, and presented together with the segmented version showing the regions, see figure 4.

Applications The demo on the web provides retrieval from a collection of 10000 Corel stock photos.

6 CANDID (Comparison Algorithm for Navigating Digital Image Databases)

Developer Computer Research and Applications Group, Los Alamos National Laboratory, USA.

URL `http://public.lanl.gov/kelly/CANDID/index.shtml`.

References [KCH95].

Features Each image is represented by a signature consisting of a weighted sum of Gaussian functions. Color, texture, and shape features are determined at every pixel, but no details are given about the exact characteristics of these features. The feature vectors of all pixels together form a point set in higher-dimensional space. On the basis of the k-means algorithms, clusters are formed. A mean vector and covariance matrix are computed for each cluster, and the associated Gaussian distribution is weighted by the number of elements in the corresponding cluster. The distribution of the feature vectors is now approximated by the weighted sum of the Gaussian distributions.

Querying The user provides a query image.

Matching The dissimilarity between two image signatures is based on the normalized Euclidean distance or the inner product of two signatures.

Result presentation Each pixel is assigned to one cluster, and each associated Gaussian distribution makes some contribution to the dissimilarity measure. In order to show which parts of the images contribute to the match, each pixel is highlighted depending on the contribution made to the similarity measure.

Applications CANDID is used in the retrieval of pulmonary CT images, and multispectral Landsat satellite images.

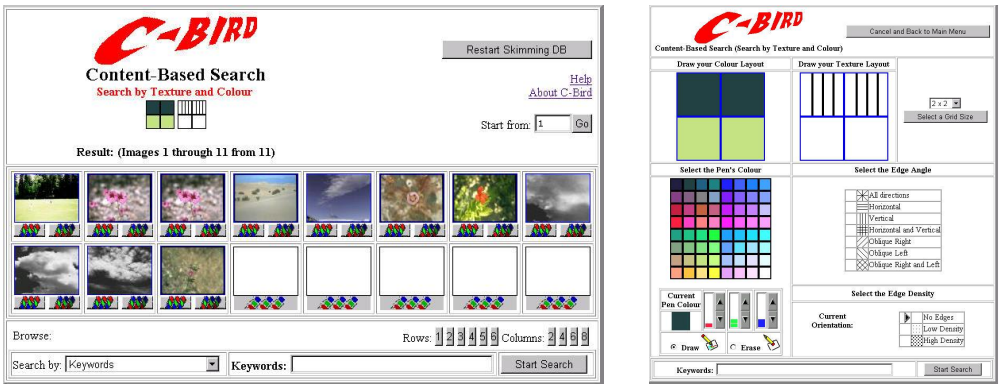

Figure 5: C-bird.

7 C-bird (Content-Based Image Retrieval from Digital libraries)

Developer School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada.

URL http://jupiter.cs.sfu.ca/cbird/

References [LZY98], [LZT99].

Features For each collected image, a feature descriptor and a layout descriptor are computed. A feature descriptor is a set of four vectors: a color vector, a most frequent color (MFC) vector, a most frequent orientation (MFO) vector, and a chromaticity vector. A 512-bin RGB histogram is stored in the color vector. The centroids of the regions associated with the 5 most frequent colors form the MFC vector and the centroids of regions of the 5 most frequent edge orientations form the MFO vector. The 36-dimensional chromaticity vector is computed as follows: first, a normalization of each RGB channel is made to obtain illumination invariance, then the 3D color histogram is replaced by a 2D chromaticity histogram. Treating this chromaticity histogram as an image, first a wavelet-based image reduction is applied, then the Discrete Cosine Transform coefficient matrix is built. The chromaticity vector is made of the 36 values of the upper left corner of the DCT matrix. For search by object model, some geometric data such as the area, the centroid and the eccentricity are computed from color regions associated with each of the MFCs.

The layout descriptor contains a color layout vector and an edge layout vector. To construct these vectors the image is divided into 64 cells, and for each cell the most frequent colors and the number of edges for each orientation are determined.

Also, for images at half and quarter resolution, a feature descriptor like the one described above is stored.

Querying The user is presented a grid of consecutive images from the database starting at a random position. To start a query by color histogram or color similarity with illuminance invariance, one of the buttons under the selected query image is pressed (see figure 5, left). For a query by color or texture layout, grids are presented for drawing color, texture density and edge orientation layout (see figure 5, right). For a query by color percentage, 5 colors and their percentages are indicated by the user. For a query by object model, the user browses through a selection of query images and makes a choice.

Matching The distance between two chromaticity vectors in an illumination invariance color query is the L_2 distance. Texture orientation histograms, as well as color histograms for the full image, are matched by histogram intersection.

The first step in a query by object model is a color localization: color regions for each MFC are extracted and for each region, some geometric data such as the area, the centroid and the eccentricity are computed. After selecting the images in the database that share a number of color regions with the query image, a number of vectors are produced by connecting the centroid of the first MFC region with the centroids of the other MFCs. Analyzing the length of these vectors and

the angles between them, a hypothesis regarding the existence of an object at a certain scale and orientation (the difference of angles between centroids of the regions corresponding to the MFCs in the query and database image) is made. This hypothesis is tested in a second step by comparing the texture histogram for each pair of matching regions in the two images. The 2D texture histogram measures orientation (the gradient direction of the edge pixels) and edge separation from the grey level image. Finally, if there is sufficient similarity in their texture between the query object and the area in the database image where the supposed similar object was identified, a shape verification based on the Generalized Hough Transform is performed [Bal81].

Result presentation The user can choose the number of rows and columns of the displayed images grid. By clicking on a thumbnail image the user can see some color and texture characteristics of the image (color percentage and layout, texture layout).

8 Chabot

Developer Department of Computer Science, University of California, Berkeley, CA, USA.

URL http://http.cs.berkeley.edu/~ginger/chabot.html. Chabot has evolved into Cypress (which, surprisingly, seems not to have inherited content based query capability). For a demo of Cypress, see http://elib.cs.berkeley.edu/photos/.

References [OS95].

Features One of the early systems, Chabot aimed at combining text based descriptions with image analysis in retrieving images from a collection of photographs of the California Department of Water Resources. The system made use of an existing text description database of the collection, adding other types of textual information for querying such as the shooting date, the picture location, the perspective of the photo. For each image a color histogram containing only 20 bins is computed.

Querying The user is presented with a list of search criteria (such as keywords, photographer, film format, shooting date, perspective, location, and colors). The color criterion offers limited options for the user to choose from, such as 'mostly red' or 'some yellow'. The user has the possibility to define concepts, which are combinations of search criteria that the concept satisfies. For example, the concept of 'sunset' is defined as a a combination of keyword ('sunset') and color ('mostly red' or 'mostly orange') criteria.

Matching To match a 'mostly ...' color criterion, more than 50% of the pixels in an image must have the requested color. For the 'some ...' color criterion, one or two of the 20 colors in the histogram must be qualified as the requested color.

Indexing The images and associated data are stored in the database POSTGRES, developed at the University of California, Berkely.

Result presentation Images are shown without specific order.

Applications The database contains 11643 images of California natural resources.

9 CBVQ (Content-Based Visual Query)

Developer Image and Advanced Television Lab, Columbia university, NY.

URL http://maya.ctr.columbia.edu:8088/cbvq/.

References [SC95].

Features The first in a series of systems (which include VisualSEEk, SaFE and WebSEEk) developed by the same team at the Columbia University, CBVQ is the only one that enables queries by texture. First, for each pixel in the image a 9-dimensional texture vector is computed. The components of this vector are the magnitudes of the output of a Haar wavelet filter bank with three iterations on the low frequency. Next, by thresholding the output of each filter, 9 binary images are produced, one for each spatial-frequency subband in the wavelet decomposition. The image pixels are merged into regions of homogeneous texture by using non-linear filtering on each binary image to reduce the noise in the texture content, followed by a sequential labeling algorithm

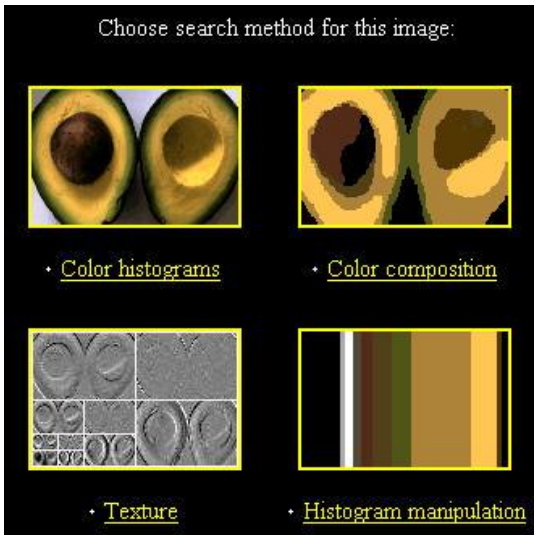

Figure 6: CBVQ.

and then a reconstruction of the image by overlapping regions from the binary subband images. Each region is represented by a 9-dimensional binary vector, where each 1 represents a high level of spatial-frequency energy within a particular subband. For each texture region detected, spatial information is also extracted, including region centroid, area, and the dimensions of the minimum bounding rectangle. For each database image, a global color histogram is also computed, and a color segmentation of the image is performed (see VisualSEEk).

Querying The system allows queries by example (the user can select one of the displayed random images or gives the URL address of any image on the Web) and direct queries (by constructing a color histogram). After selecting the query image in a query by example, the user chooses one of the available search methods (see figure 6): color histogram, color composition, texture and histogram manipulation (the query image histogram can be used to construct a new query histogram).

Matching Matching by texture regions is most probably done in the same way as color region matching is performed in VisualSEEk.

Indexing See VisualSEEk.

Result presentation The best 20 matches are presented to the user in decreasing similarity order, together with the matching score.

Relevance feedback Any of the retrieved images can be chosen as the new query image.

10 DrawSearch

Developer Department of Electrical and Electronic Engineering, Technical University of Bari, Italy.

URL A demo is available at `http://deecom03.poliba.it/DrawSearch/DrawSearch.html`.

References [SMM99].

Features The features used for querying are color, texture and shape. The image is divided into 4×4 subimages and within each subimage, the average color is computed. This results in a 48-length color vector. Shape is represented by means of Fourier descriptors. It is unclear how the contour-based segmentation of the image is done. The components of the shape descriptor are the real part of the lower 100 Fourier coefficients. Texture-based image segmentation makes use of Gaussian Markovian Random Fields.

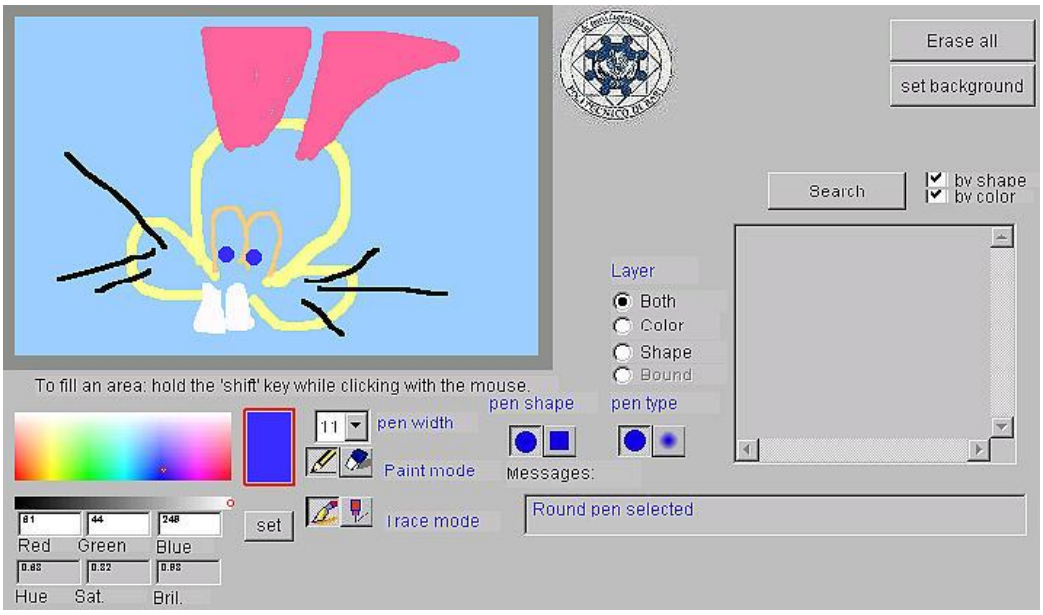

Figure 7: DrawSearch.

Querying The system has two user interfaces: one that allows queries by color and shape, and the other one allowing queries by texture. In the first subsystem, the user draws a sketch on a canvas, selecting color attributes for the lines and/or closed regions (see figure 7). In the texture retrieval subsystem, a query is posed by selecting a texture area within a database image.

Matching In the color/shape subsystem, the similarity between two feature vectors is given by the cosine metric. The similarity score between a query and a database image is calculated as a weighted sum of the distances between the two color vectors and shape descriptors.

Result presentation Images are displayed in decreasing similarity order.

Relevance feedback In the retrieval by texture subsystem, the user can improve retrieval results by selecting relevant images from the displayed query results. The other subsystem allows both relevant and nonrelevant images to be marked by the user when reiterating the query. The new query vector is computed by combining the feature vector of the original query Q with the feature vectors of relevant and nonrelevant images: $Q^{k+1} = Q^k + \delta \sum_{i=1}^{n_{rel}} X_i - \epsilon \sum_{j=1}^{n_{notrel}} Y_j$, where δ, ϵ are positive real weights, X_i is the feature vector of the ith relevant image, and Y_j of the jth nonrelevant image.

11 Excalibur Visual RetrievalWare

The Visual RetrievalWare is a software developers kit for building applications for manipulating digital image files and their visual content. The toolkit contains C++ and Java API's for image processing, feature extraction, indexing and content-based retrieval. It also includes sample programs which might be used directly or can serve as templates for building more complex applications. One of these sample programs is the CST (Color, Shape, and Texture) demo.

Developer Excalibur Technologies.

URL http://vrw.excalib.com/. A demo is available at `http://vrw.excalib.com:8015/cst`.

Features The CST demo allows queries by example based on HSV color histograms, relative orientation, curvature and contrast of lines in the image, and texture attributes, that measure the flow and roughness in the image.

Querying The user first defines the desired visual similarity by specifying the relative importance of the above image attributes, and then selects one of the displayed images as query, see figure 8.

Figure 8: Excalibur: result of querying with upper left image based on shape alone.

Result presentation The images are shown without an explicit ordering.

Applications The software has been used in the Image Surfer system, which was used by the Yahoo! and Infoseek WWW search engines.

12 FIR (Formula Image Retrieval)

Developer Developed by Fraunhofer Institute for Computer Graphics, Darmstadt, Germany, in association with Txt Ingegneria Informatica S.P.A. (Italy), Giunti Multimedia Srl (Italy), Epsilon Software (Greece), and Kino TV & Movie Productions S.A. (Greece), as part of the Esprit IV project FORMULA.

URL http://www.igd.fhg.de/igd-a7/projects/formula/formula_e.html

References [Vol97].

Features Basic shape and color information of an image are represented using a multiresolution wavelet decomposition. A preprocessing step include a rescaling of the image to a square thumbnail of 128×128 pixels and a transformation of the thumbnail colors from the RGB space to Luv space. Next, a non-standard two-dimensional Haar wavelet transform is performed on each color channel individually, followed by a truncation of the resulting coefficients. Only the coefficients larger than a threshold are retained and after rounding to integer values they are stored in a feature vector.

Querying A query is submitted by giving an example image.

Matching The distance measure between two feature vectors is a weighted Euclidean distance, with different weights for each of the resolution levels in the wavelet decomposition.

13 FOCUS (Fast Object Color-based Query System)

Developer Department of Computer Science, University of Massachusetts, Amherst, MA.

URL http://wagga.cs.umass.edu/~mdas/color_proj.html. A demo of the system is available at http://cowarie.cs.umass.edu/~colordemo/mdas/demo1/phase0.html.

References [DRD97].

Querying The user can select as query one of the displayed template images, or create a new template by marking a subimage which contains the region of interest.

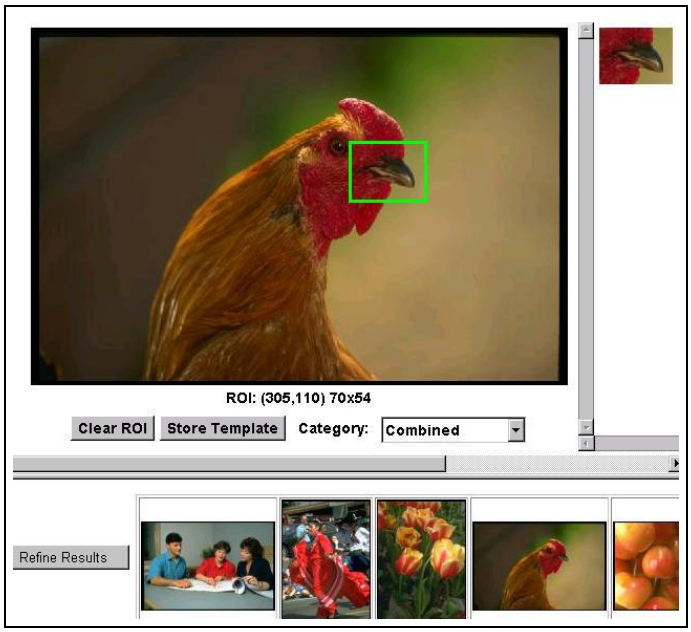

Figure 9: Focus. Result of querying with the drawn region of interest.

Features Each image is divided in cells of 100×100 pixels and for each cell a color histogram in the HSV space, coarsely quantized along the saturation and value axes (64×10×10), is computed. The peaks of all local histograms are determined and combined in a list of unique peaks for the whole image by merging multiple copies of the same peak. Also, a frequency table is constructed which, for each color in the HSV space, gives the number of images that have a peak of that color.

The spatial relationships between color regions are represented by means of a spatial proximity graph (SPG) constructed in two phases. First an intermediate SPG is generated, with one node corresponding to each color peak computed for the image cells. Two nodes in this graph are connected if their corresponding peaks are located in the same cell or are located in neighboring cells and have the same color. This graph is then simplified, by unifying all connected nodes of the same color in a single node, and stored using an adjacency matrix representation.

For the query image, a global color histogram is computed and color region relationships are determined at pixel level.

Matching The peaks of a query image are subjected to approximate range queries in the increasing order of their corresponding entries in the frequency table. From the resulting lists, the set of images which have peaks matching all query peaks are determined. For the images in this set, a matching score is computed as the the sum of the L_1 distances between each query peak and the matched candidate peak.

To match the SPG of the query image with that of a candidate image, first the candidate SPG is reduced by removing any node whose corresponding peak does not match a query peak. Then it is checked if the query graph appears as a subgraph in the candidate SPG.

Indexing The color peaks of the database images are stored in a B^+ tree [KS91] sorted with hue as the primary key, followed by saturation and value.

Result presentation When the user submits a query by clicking on an image, the images are retrieved using the first phase of matching (the images displayed are the images that have all the colors of the query image). By clicking on the 'Refine Results' button, the retrieved images are subjected to the second phase of matching, where the spatial relationships of the matched color regions is analyzed in order to detect a query object in a candidate image.

Applications The database consists of 400 advertisements and 800 color natural images.

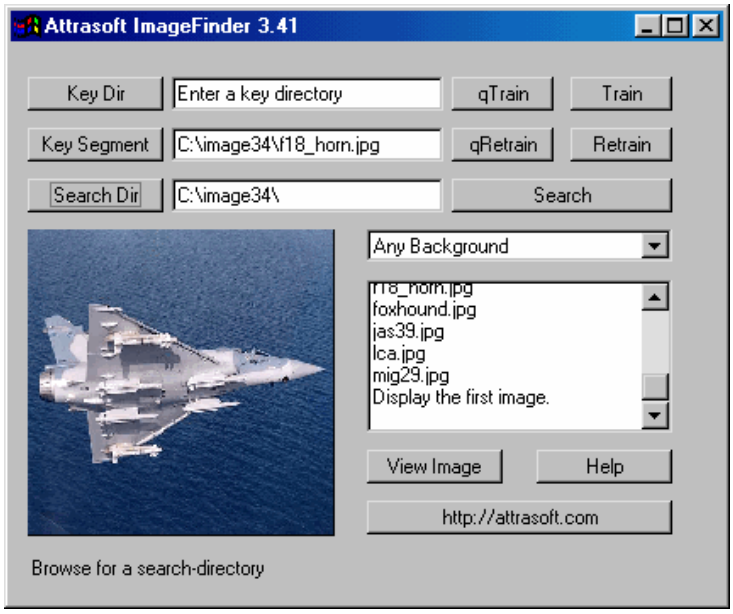

Figure 10: ImageFinder training.

14 ImageFinder

Developer Attrasoft Inc.

URL http://attrasoft.com/abm3_4.html.

Features It remains unclear on what features the matching is done.

Querying The querying is done by giving a query image.

Matching Matching is done using a Boltzman Machine, a special kind of probabilistic artificial neural network. The network must first be trained on a set of images. This is done by just entering the training images (key images), there is no control over the features on which to train the network, see figure 10.

Result presentation The result is an HTML page with a list of links to the images found.

15 ImageMiner

Developer Technologie-Zentrum Informatik, Univerity of Bremen, Germany.

URL http://www.tzi.de/bv/ImageMinerhtml/.

References [KRA+97].

Features ImageMiner generates content descriptions of images based on color, texture and contours. A color segmentation of the image is done by first computing a color histogram for all grid elements of homogenous size that the image is divided into and then grouping the grid elements according to their most frequent color in the histogram. Bounding rectangles are determined for the grouped grids. For each rectangle, attributes like size, position (both relative to the grid size), associated color and color density (the ratio of the size of color rectangle to the amount of grid elements with that color) are stored.

The texture segmentation is similarly given by the bounding rectangles of the grids grouped according to their similar texture. For representing texture, a statistical approach is adopted. For every grid element, the cooccurrence matrix P is calculated for four orientations (0°, 45°, 90° and 135°). This is an $N_g \times N_g$ matrix (with N_g representing the number of distinct gray levels in the quantized image) where P_{ij} is defined as the joint probability that a pair of neighboring pixels, one with gray-level i and the other with gray-level j, occur in the image under the four specific angular

relationships. From each matrix, five statistical features are determined (angular second moment, contrast, correlation, variance and entropy). The average values of these features over the four orientations are the inputs of a neural network trained with a backpropagation algorithm. The network output neurons symbolize seven visual texture properties: bloblikeness, multiareas, planarity, coarseness, regularity, directionality and softness. For every texture rectangle determined, its size, position, the classified texture and the texture density are stored.

The contour-based shape description is obtained by first detecting the edge points using a gradient method and then connecting these points to form contours. For the closed regions detected some shape parameters are extracted (the coordinate of the middle point, the size and the bound coordinates).

The neighborhood relations of the contours and the color and texture regions are represented in a graph. Graph parsing algorithms recognize objects and generate a description with keywords and values.

Querying SQL-like queries with keywords and values can be posed.

Matching and Indexing Retrieval is performed by the IBM Search Manager text searcher.

Applications Videos are made available for retrieval with ImageMiner in two steps. First, shots are extracted from the video by using a histogram based method. More precisely, a new shot is detected when the difference in the intensity histogram of two consecutive frames exceeds a given threshold. Then, a mosaicing-technique is used to generate a single still image for each shot detected in the video sequence. This mosaiced image comprises all the information in a shot and is analyzed by the ImageMiner system as above.

16 ImageRETRO (Image RETrieval by Reduction and Overview)

Developer Department of Computer Science, University of Amsterdam, The Netherlands.

URL http://carol.wins.uva.nl/~vendrig/imageretro/. For a demo of the system, look at http://python.wins.uva.nl:6022/.

References [VWS99].

Querying The user is shown a number of images (called overview), each of them being representative for a part of the image collection. By selecting one of these images, the user selects a part of the database, which is then clustered, and another set of representative images is shown to the user. By this successive filtering, the user obtains a small set of images which can be browsed manually.

Features The following features are used in clustering images: *a 15-bin hue histogram*, *the number of colors* (the number of bins with more pixels than a threshold, relative to the total number of bins in the hue histogram), *the average color* (the center of gravity of the circular hue histogram is the bin which gives the lowest average value when taken as a starting point in computing this value; the average value is given by $avg = (\sum_{j=1}^{15} h_j d_j)/h$, where h_j represents the number of pixels in bin j, d_j is the distance from this bin to the starting point in the circular hue histogram, and $h = \sum_{j=1}^{15} h_j$; the average color is the average value of the center of gravity divided by 15), *the color variation* (the circular hue histogram is first linearized by using the center of gravity as the center of the new histogram; the standard deviation of this histogram normalized by the maximal possible standard deviation gives the color variation), *the grayness* (share of gray valued pixels in the total number of pixels), *the average gray value* (average value of the gray value histogram, see average color), *the average saturation* (the sum of saturation values for each pixel divided by the total number of pixels), *the background proportion* (share of the background in the total number of pixels, where the supposed background is the bin with the highest frequency in the hue or the gray value histogram), *the number of regions* (a region is a collection of adjacent pixels with the same color hue or gray value), *the number of holes* (relative to the number of regions, where a hole is a region surrounded entirely by another region). For all the images in the database, the color features described above are computed off-line and stored in the database.

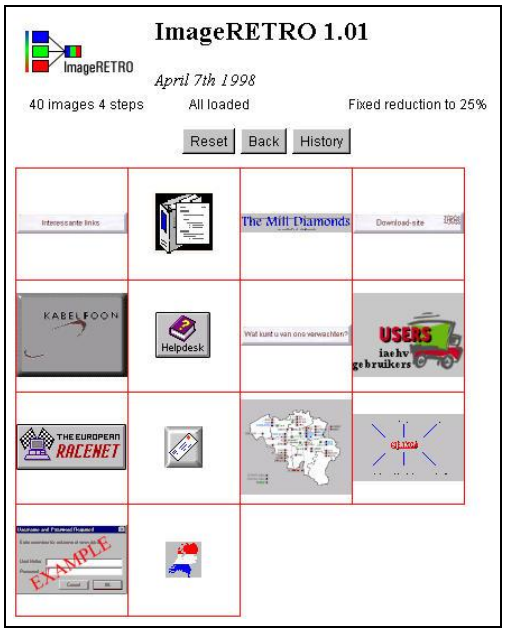

Figure 11: ImageRETRO.

Matching The ranking of the image clusters, employed in the relevance feedback, is straightforward when based on one of the scalar color features. No histogram distance measure is mentioned.

Relevance feedback Let I_s be the image set after s reductions (filterings) and let F denote the set of 10 color features described. The image set is clustered based on an automatically selected feature subset F_s of F. The images from I_s are ranked independently for each feature in F_s, and each such ranking is divided into 4 clusters (corresponding to a reduction of 25%) and each cluster centroid is chosen as the cluster representative. The union of these representatives for all rankings forms the representative set of I_s, which will be shown to the user for the next reduction. The choice of feature subset F_s at stage s in the retrieval process is based on statistical analysis. For each feature, the variance of the feature values of all images is computed and F_s is made of the features with highest variances, that do not highly correlate with each other.

Applications The system was tested with a database of 10000 images, collected from the Web.

17 ImageRover

Developer Department of Computer Science, Boston University, MA.

URL http://www.cs.bu.edu/groups/ivc/ImageRover/. For a demo of the system, look at http://www.cs.bu.edu/groups/ivc/ImageRover/demo.html.

References [STC97], [TCS97].

Features The features used for querying are color and texture orientation. The system computes distributions of color and orientation over 6 subimages (the whole image and 5 subregions: the central and corner regions). The result is an image index vector made of 2×6 subvectors. This dimension is subject to a reduction via a principal component analysis (PCA) for each of the subvector spaces. Image color histograms are computed in the CIE Luv color space and each histogram quantizes the color space into 64 bins. The texture direction distribution is calculated using steerable pyramids. At each of the four levels of the pyramid, texture direction and strength for each pixel is calculated resulting in an orientation histogram, quantized to 16 bins.

Querying The user starts a query session by specifying a set of keywords related to the desired images. From the set of images displayed, the user finds and marks one or more images which are similar to what she/he is looking for. Apart from selecting relevant images, the user can deselect

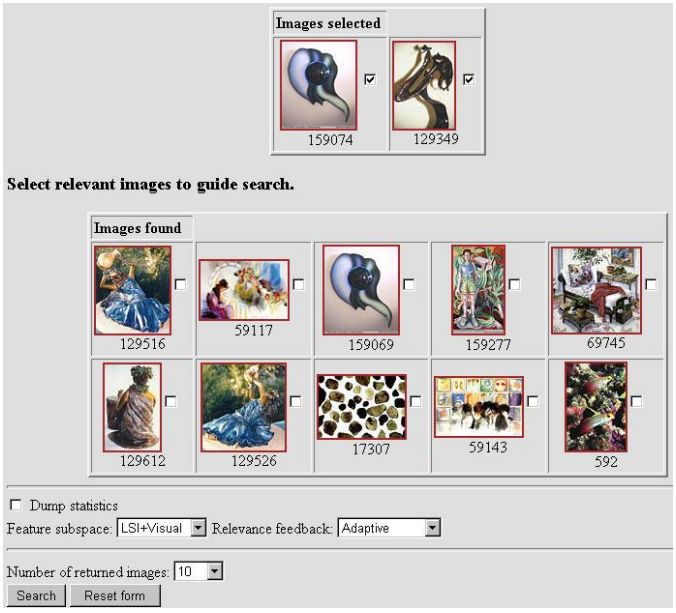

Figure 12: ImageRover.

one or more of the query images before reiterating a new query. There is no limit to the number of iterations in providing relevance feedback, nor in the number of example images.

Matching Based on relevance feedback from the user, the system selects the appropriate $\tilde{L}_m$ normalized Minkowski metric each time a query is made. The normalization factor μ_m^i, employed by the normalized distance $\tilde{L}_m(x_i, y_i) = L_m(x_i, y_i)/\mu_m^i$, with x_i and y_i being two subvectors of the image index vectors X and Y, is the expected value over the entire database: $\mu_m^i = E[\tilde{L}_m(x_i, y_i)]$. It is allowed to use metrics of different orders m_i for each of the image index subvectors. Thus, if S is a set of relevant images indicated by the user, the appropriate value for m of the i-th subvector is chosen so as to minimize the distance between relevant images: $m_i = arg\ \min_m E[\tilde{L}_m(p_i, q_i)]$, over all $P, Q \in S$.

Then the k-nearest neighbor search of the image index vector uses the following weighted distance metric: $d(X,Y) = (w_1, ..., w_n)(\tilde{L}_{m_1}(x_1, y_1), \ldots, \tilde{L}_{m_n}(x_n, y_n))^T$ where X, Y are the image index vectors and w_i are relevance weights.

Indexing An optimized kD tree is used and the search algorithm in this structure employs an approximation factor. The user can select through the search accuracy button one of the three values for this approximation factor.

Result presentation Images similar to the query images are shown in decreasing similarity order. Their number is chosen by the user. Each displayed thumbnail image is a hypertext link to the original image, allowing the user to retrieve the desired image from its home WWW site.

Relevance feedback The user indicates the more relevant images, after which the system selects the appropriate $\tilde{L}_m$, see above.

Applications Image Rover is an image search engine for the WWW.

18 ImageScape

Developer Department of Computer Science, Leiden University, The Netherlands.

URL http://www.wi.leidenuniv.nl/home/lim/image.scape.html. A demo of the system is available at http://ind134a.wi.leidenuniv.nl:2001/new2/imagesearch.demo.html.

References [LLH97], [BL99].

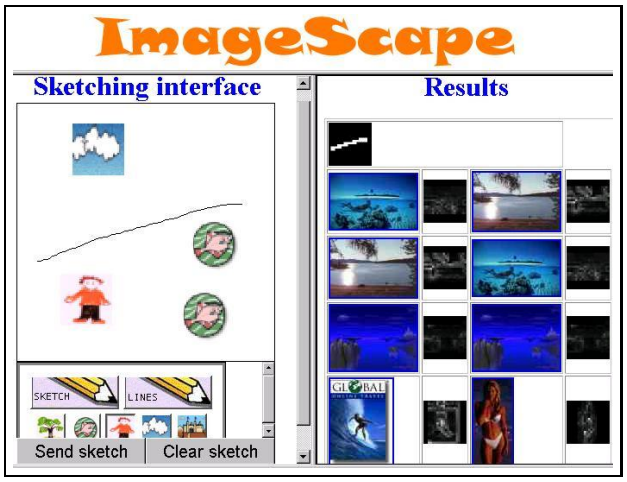

Figure 13: ImageScape query built with icons.

Querying Using the sketch interface, the user can draw an outline of the desired image. For semantic querying, the user brings icons on a canvas that represent the objects/concepts he is looking for, at the desired position in the image. Examples of object/concept categories include human faces, stone or sand, water, sky, tree or grass, points and lines (see figure 13).

Features Edge maps of the images collected by Web crawlers are obtained using the Sobel operator and a Gaussian blurring filter. A frequency histogram of the 3×3 binary pixel patterns occurring in the edge image, which is called trigram vector, is computed for all images. This vector is subjected to a dimensionality reduction using a band-pass filter (see LCPD below). Various other features, used in object matching, are taken at pixel level: color, Laplacian, gradient magnitude, local binary patterns, invariant moments and Fourier descriptors.

Matching The first step of the object matching process uses the L_1 distance on the trigram vectors to retrieve the top 1% matches from the entire database. Among these, 20 matches are selected in a second step, a 20×20 template matching, using the most informative pixels to minimize the misdetection rate. These pixels are found as follows. For each object, a large set of positive and negative examples are used in finding the set of 256 pixels with the greatest discriminatory power, by maximazing the Kullback relative information combined with a Markov random field.

19 Jacob Just A COntent Based query system for video databases

Developer Computer Science & Artificial Intelligence Lab, University of Palermo, Italy.

URL http://www.csai.unipa.it:80/research/projects/jacob/. A demo is available at site http://www.csai.unipa.it:80/research/projects/jacob/jacob_demos.html.

References [CA96].

Features The system makes queries based on color and texture features. Color is represented by means of a histogram in the RGB space. Texture features used are two measures extracted from the grey-level cooccurrence matrix, the maximum probability $max_{m,n} f(m,n,r,\theta)$, and the uniformity $\sum_m \sum_n (f(m,n,r,\theta))^2$. An 8-dimensional vector is obtained by computing the joint probability $f(m,n,r,\theta)$ for distance $r = 1$ and orientation $\theta = -45°, 0°, 45°$, and $90°$. Another vector is composed of edge density measures (the fraction of edge pixels in the total number of pixels in the image) along four directions, $-45°, 0°, 45°$, and $90°$.

Querying The queries may be direct or by example. A direct query is made by inserting a few values representing the color histogram and/or the texture features. For a query by example, the user must give an image.

Matching Two color histograms are compared using the following distance measure: $d(H_1, H_2) = \sum_{y \in S} \sum_{x \in I(y)} |H_1(y) - H_2(x)| s(y,x)$ where H_1, H_2 are the histograms, S is the reduced RGB

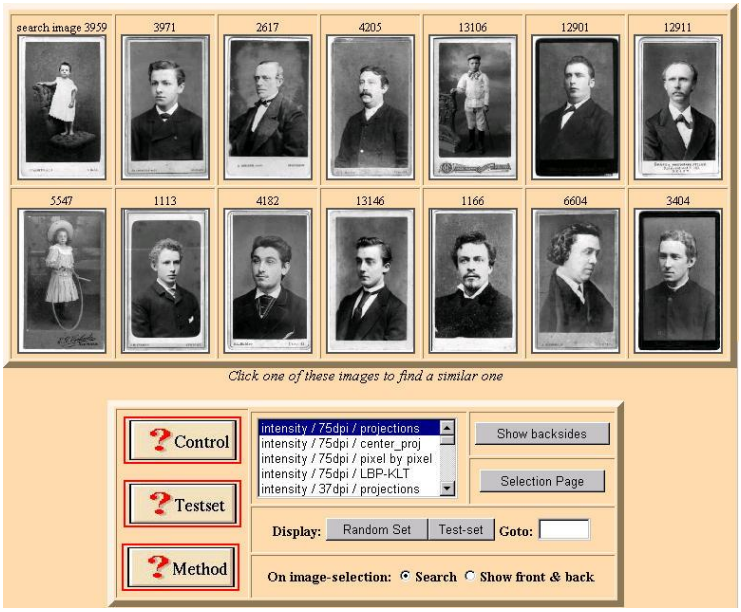

Figure 14: LCPD.

space, $I(y)$ is a small interval centered on the point y and s is a color similarity function. The distance measure used for the texture feature vectors is not mentioned. When performing a query, the user choose a value between 0 and 1 to indicate the relative importance of a feature with respect to the other one. As a result the two distances computed are weighted in a global similarity measure.

Result presentation The system returns the best matching frames in a similarity order. The number of returned frames is chosen by the user.

Applications The matching of images is used for querying video databases.

20 LCPD (Leiden 19th Century Portrait Database)

Developer Department of Computer Science, Leiden University, The Netherlands.

URL http://ind156b.wi.leidenuniv.nl:2000/. A demo of the system is available at site http://ind156b.wi.leidenuniv.nl:2000/cgi-bin/image_demo6.0.pl.

References [LHD96].

Features The user has several choices in selecting the feature vector, the pixel value domain used for computing this vector, and the resolution level. There are three options for the pixel domain: the intensity image, the gradient image (obtained by Sobel operators) and the thresholded gradient image. One feature vector is the horizontal/vertical projection vector. For an image with $m \times n$ pixels, this vector has $m + n$ components computed as the average of the row/column pixel values in the selected space. A second feature is the trigram vector, a frequency histogram of the 3×3 binary pixels patterns in the thresholded gradient image. This 512-length vector can be subjected to a dimensionality reduction and to a component weighting scheme (low weights are applied on either end of the sorted frequency range). A way of reducing the length of the trigram vector is by forming groups of rotation, mirroring and/or intensity invariant binary patterns. By using the RM (Rotation, Mirroring) group, the dimension of the feature vector is reduced to 102 and by forming RIM (Rotation, Intensity, Mirroring) groups to 51. Another method used consists of suppressing the contribution of the black and white patterns (which are among the most common patterns) and the rare patterns. This is called a bandpass filter. A Karhunen-Loeve Transform can also be used for feature vector length reduction.

A similar vector can be constructed in the intensity space. In this case, the feature vector is computed by first thresholding the 8 neighbors of each pixel with its intensity value and counting the number of occurrences of each of the 2^7 possible patterns of these 8 pixels (the center pixel is not part of the pattern). This results in an 256-length local binary pattern vector (LBP), which can also be subjected to dimensionality reduction.

Querying Querying is done by example. The user first selects a search method (different combinations of pixel domain, feature vector, dimensionality reduction scheme employed and resolution) from the displayed list.

Matching The similarity between two feature vectors (projections, trigram or LBP vectors) is given by the L_1 distance. If two images have different sizes, the similarity score between two projection vectors is given by the minimum L_1 distance over all possible x-/y-translations of one image over the other. Another similarity measure is given by the magnitude of the the average pixel to pixel difference in the intensity or gradient space.

Indexing The POSTGRES database stores for each image the best matches for 30 search methods.

Result presentation Retrieved images are presented in decreasing similarity order, see figure 14).

Applications The system is used to find near copies of Dutch carte de visite studio portraits from a database of 16505 images of photographs taken from 1860 till 1914.

Performance http://ind156b.wi.leidenuniv.nl:2000/cgi-bin/performance/m_page1a.pl provides information about performance.

21 MARS (Multimedia Analysis and Retrieval System)

Developer Department of Computer Science, University of Illinois at Urbana-Champaign, further developed at Department of Information and Computer Science, University of California at Irvine, CA.

URL http://www-db.ics.uci.edu/pages/research/mars.shtml. Demos are available at location http://www-db.ics.uci.edu/pages/demos/index.shtml.

References [ORC+97].

Features The system supports queries on combinations of low-level features (color, texture, shape) and textual descriptions. Color is represented using a 2D histogram over the HS coordinates of the HSV space. Texture is represented by two histograms, one measuring the coarseness and the other one the directionality of the image, and one scalar defining the contrast. In order to extract the color/texture layout, the image is divided into 5×5 subimages. For each subimage a color histogram is computed. For the texture of a subimage, a vector based on wavelet coefficients is used. The object in an image is segmented out in two phases. First, a k-means clustering method in the color-texture space is applied, then the regions detected are grouped by an attraction based method. This consists of choosing a number of attractor regions and associating each region with the attractor that has the largest attraction to it. The attraction between two regions, i and j, is defined as $F_{ij} = M_i M_j / d_{ij}^2$, where M_i, M_j are the sizes of the two regions and d_{ij} is the Euclidean distance between the two regions in the spatial-color-texture space. In the MARS system, five attractors are used: one for each corner of the image (background attractors) and one in the center of the image (the objects attractor). This is consistent with the fact that their database consists of images of single objects. The shape of the boundary of the extracted object is represented by means of Fourier Descriptors (FD).

Querying Complex queries can be formulated using boolean operators. The desired features can be specified either by example (pointing an image database that has such a property) or direct (for example, by choosing colors from a palette or textures from an available set of patterns).

Matching The similarity distance between two color histograms is computed by histogram intersection. The similarity between two textures of the whole image is determined by a weighted sum of the Euclidean distance between contrasts and the histogram intersection distances of the other two components, after a normalization of the three similarities. For computing the texture

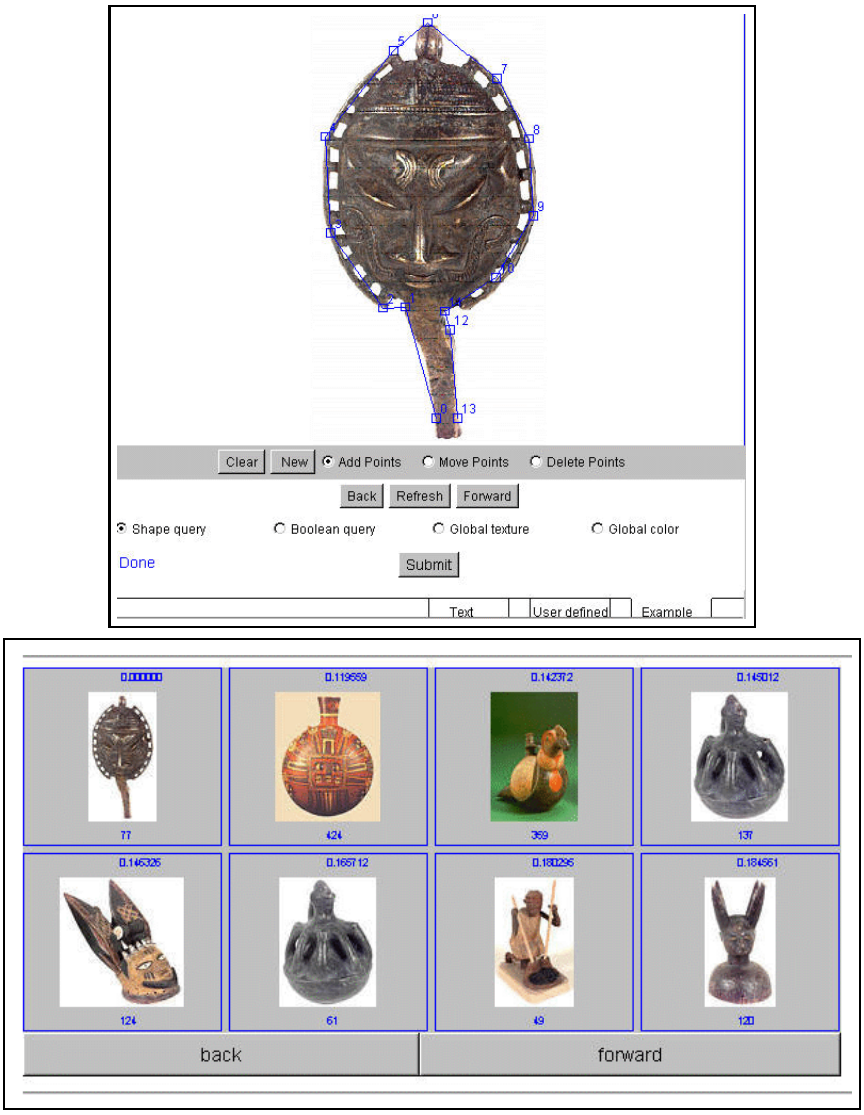

Figure 15: Mars shape query with drawn polygon.

similarity between two corresponding subimages, the Euclidean distance between the vector representations is used. A weighted sum of the 5×5 color/texture similarities is used to compute the color/texture layout distance between two images. The similarity measure between two FD shape representations is a weighted sum of the standard deviations of $ratio(k) = M_2(k)/M_1(k)$ and $shift(k) = \theta_2(k) - \theta_1(k) - \psi$, $k = -N_c, \ldots, N_c$, where $M_i(k)$ and $\theta_i(k)$ are the magnitude and the phase angle of the FD coefficients, ψ is the difference of the major axis orientations of the two shapes and N_c is the number of FD coefficients.

Each query has a query tree associated. In a query tree, the leaves represent the feature vectors (the terms of the boolean expression defining the query) while the internal nodes correspond to boolean operators or more complex terms indicating a query by object. Individual queries on each of the query terms are made. The tree is evaluated bottom-up: each internal node receives from each child a list of ranked images and combines these lists, after a normalization process, according to the weights on the parent-child links.

Indexing There is no information about indexing data structures used for queries. A new version of the system, WebMARS, is developed where the feature vectors are indexed using hybrid trees,

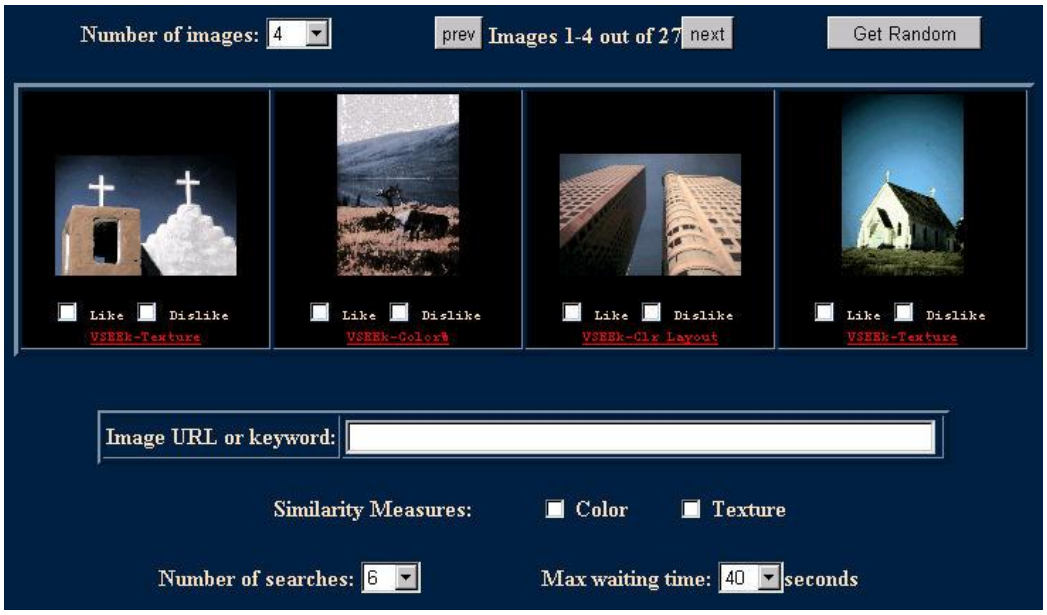

Figure 16: MetaSEEk results and relevance feedback.

which combine aspects of several indexing trees.

Result presentation Images are listed in order of decreasing similarity.

Relevance feedback Initially, the weights of the edges in the query tree are equal for the children of the same parent and their sum is 1. Based on the relevant images chosen by the user from the query result list, a tree reweighting process takes place.

Applications The database consists of images of ancient African artifacts from the Fowler Museum of Cultural History at UCLA.

22 MetaSEEk

Developer Image and Advanced Television Lab, Columbia University, NY, USA.

URL http://www.ctr.columbia.edu/metaseek/.

References [BBC98].

Features Content-based retrieval can be done on the basis of color and texture. MetaSEEk has a color or texture matching and indexing of its own to cluster query images in a local performance database, used to select target serach engines. The actual image matching is forwarded to QBIC, VIR Image Engine, WebSEEk, and the VisualSEEk retrieval engines.

Querying The user can select a category, provide a key word, provide a URL of an image, or select a shown image. Upon receiving a query, the dispatcher consults the performance database at the MetaSEEk site. This database contains performance scores of past query successes and failures of each supported search option. This is used to select the target search engines to be queried.

Indexing Query images in the performance database are clustered into several classes on the basis of color, textures, and the combination of both. When the user issues a query with a new image for which no performance scores are available, the system downloads the image, and matches it to the corresponding clusters in order to obtain a list of the most similar clusters. Selected images from the few closest clusters are presented to the user, who can choose one. In this way, new queries are related to the performance of past ones for which performance information is available.

Matching Color and texture are extracted locally by MetaSEEk for the clustering. Color similarity is computed by calculating the color histogram of an image. Texture is computed by measuring

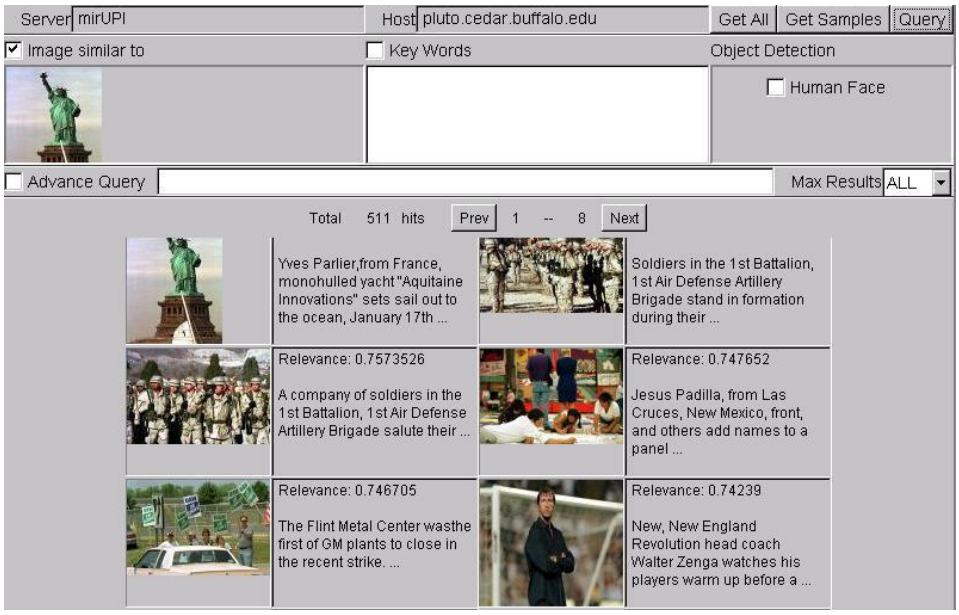

Figure 17: MIR.

the coarseness, contrast, and presence/absence of directionality of an image. The distance between two feature vectors is the Euclidean distance.

Result presentation The display component merges and ranks the results form each search option.

Relevance feedback The user can indicate to like or dislike each of the returned images, see figure 16. This information is used to modify the corresponding entries in the performance database.

23 MIR (Multimodal Information Retrieval System)

Developer Center of Excellence for Document Analysis and Recognition, University at Buffalo, NY, USA.

URL `http://www.cedar.buffalo.edu/MMIR/`. A demo of the system is available at webaddress `http://www.cedar.Buffalo.EDU/MMIR/demo/index.html`.

References [Sri95], [SZR00].

Features The MIR system combines various techniques from text processing and image processing in an attempt to derive semantic descriptions of images. By employing natural language processing (NLP) techniques in analyzing image captions, information about the picture's content is derived. This includes whether there are people in the picture, their names, location and time of the photograph, spatial relationships of the people in the image as well as other visual characteristics (attributes that can assist in face recognition such as gender, hair color, beards, mustaches and glasses). Also, statistical text indexing techniques are used in capturing the general context of the image (such as indoor versus outdoor). The pictures in which the presence of people is detected by NLP, are subjected to a face detection to verify this hypothesis. The face detection algorithm uses a three-contour model representing hair line, and left and right outlines of the face. After extracting edges at different image resolutions by means of a wavelet transform, a graph-matching generates possible face candidates in the image. After the face detection, the face areas are cropped out, and for the rest of the image a color histogram is computed. For the scenery pictures (where no face is detected in the NLP analysis) a color correlation histogram is computed. If $\{i, j, k\}$ is a color triple in the $2 \times 2 \times 2$ quantized HVS space, then h^l_{ijk} denotes the number of occurrences of three pixels of these colors as the vertices of a isosceles right triangle with the smaller sides of length l. The color correlation histogram f^l_{ijk} is $h^l_{ijk}/(4 * h_i)$, where h_i is the i^{th} bin of the traditional color histogram.

Querying A query formulation can include a text string, an image, and a topic (selected from a predefined set containing sports, politics, entertainment etc.). The user can also indicate the relative importance of text versus image content, as well as of background versus foreground in an image containing people (the foreground).

Matching The similarity between a query and a database image is a weighted sum of the similarities between different sources of information (text-based and content-based) extracted from the image. Two color correlation histograms are matched using the Euclidean distance.

Applications The system retrieves pictures of people and/or similar scenery in various contexts. Three databases were used in testing the system. The first one consisted of approximately 5000 images with accompanying text provided by United Press International. The second one provided by Kodak consists of consumer photos accompanied by speech annotations. The third one consists of multimodal documents downloaded from the Web.

24 NETRA

Developer Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA.

URL http://maya.ece.ucsb.edu/Netra/. A demo of the system is available at webaddress http://maya.ece.ucsb.edu/Netra/netra.html.

References [Ma97], [MM99].

Features Images in the database are segmented into regions of homogeneous color. Of those regions, the following features are extracted: color, texture, shape, and spatial location.

On the basis of a training set of images, the RGB color space is quantized, and represented by a color codebook of 256 colors, the centroids of the quantization cells. The colors of an image region are also quantized, giving a color feature vector $f_c = (c_0, p_0, \ldots, c_n, p_n)$, with c_i the index into the color code book, and p_i the fraction of that color in the region, $p_0 + \ldots + p_n = 1$. The number n is the number of colors used to represent the region, which is different for each region.

Texture is represented by a feature vector f_t containing the normalized mean and standard deviation of a series of Gabor wavelet transforms of the image: $f_t = (\mu_{0,0}, \ldots, \mu_{s,k}, \sigma_{0,0}, \ldots, \sigma_{s,k})$, with s the number of scales, and k the number of directions.

There are three feature vectors used to represent the shape of regions. The first, f_K, is based on the curvature function of the contour, giving the curvature at each point on the contour. The second, f_R is based on the centroid distance function, giving at each contour point the distance to the centroid of the region. The third, f_Z, is the complex coordinate function, representing each contour point as a complex number with real component equal to the x-coordinate, and the imaginary component equal to the y-coordinate. On 64 samples of each of these functions, the fast Fourier transform (FFT) is applied, of which the real (amplitude) component of the coefficients is used, the numbers $F_{-31}, \ldots, F_{32}$. The feature vectors are as follows: $f_K = (|F_1|, \ldots, |F_{32}|)$, $f_R = (|F_1|, \ldots, |F_3 2|)/|F_0|$, $f_Z = (|F_{-31}|, \ldots, |F_{-1}|, |F_2|, \ldots, |F_{32}|)/|F_1|$.

Querying There are 2,500 images from the Corel photo collection, organized in 25 categories, with 100 images in each category. You can select any one of them as the query image. All images in the database have been segmented into homogeneous regions. You can click on one of the regions and select one of the four image attribute color, spatial location, texture, and shape. Instead of using an image example, you can also directly specify the color and spatial location. The spatial location querying tool utilizes two bounding boxes to define the area of interest. The inner box is used to define the preferred area, and the box outside is used to constrain the objects to be within this area. Thus, if the object has any its bodies exceeding this outside box, they will not be considered.

Matching Consider two color feature vectors, f_c^A of region A, and f_c^B of region B. For each color c_i in f_c^A, the closest color c_k^B in f_c^B is found, and the distance $d(c_i^A, f_c^B)$ is calculated as the weighted Euclidean distance in RGB space: $d(c_i^A, f_c^B) = |p_i^A - p_k^B| d(c_i^A, c_k^B)$. The distance between the two color feature vectors is now $\sum_{i=0}^{n^A} d(c_i^A, f_c^B) + \sum_{i=0}^{n^B} d(c_i^B, f_c^A)$. The distance between two

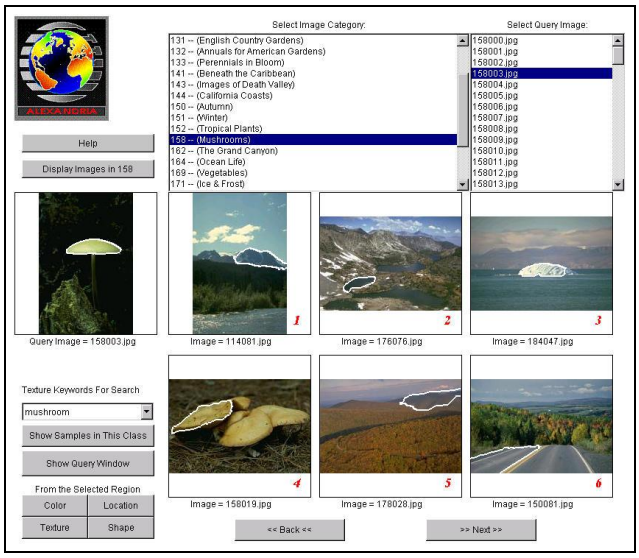

Figure 18: NETRA. Result of querying on shape with the complex description.

texture feature vectors is the L_1-distance. The distance between two shape feature vectors is the Euclidean distance.

Indexing is based on the SS-tree [WJ96]. Color, texture, and shape are indexed separately. The first feature the user specifies is used to retrieve about 100 candidates. Then this feature and the possible other features together are used to order the retrieval result.

Result presentation The matched images are linearly ordered, see figure 18.

Applications An initial prototype of NETRA is used in ADL (see above) to search on texture.

25 Photobook

Developer Vision and Modeling Group, MIT Media Laboratory, Cambridge, MA.

URL http://vismod.www.media.mit.edu/vismod/demos/photobook/index.html. A demo is available at http://vismod.www.media.mit.edu/cgi-bin/tpminka/query?vistex,,,10.

References [PPS96].

Features Photobook implements three different approaches to constructing image representations for querying purposes, each for a specific type of image content: faces, 2-D shapes and texture images. The first two representations are similar in the way that they offer a description relative to an average of a few prototypes by using the eigenvectors of a covariance matrix as an orthogonal coordinate system of the image space. First a prepocessing step is done in order to normalize the input image for position, scale and orientation. Given a set of training images, $\Gamma_1, \Gamma_2, ..., \Gamma_M$, where Γ_i is a $n \times n$ array of intensity values, their variation from the average, $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$, is given by $\Phi_i = \Gamma_i - \Psi$. This set of vectors is then subjected to the Karhunen-Loève expansion, the result being a set of M eigenvectors u_k and eigenvalues λ_k of the covariance matrix $C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T$. In representing a new image region, Γ, only $M_1 < M$ eigenvectors with the largest eigenvalues are used, thus the point in the eigenimage space corresponding to the new image is $\Omega = (\omega_1, \omega_2, ..., \omega_{M_1})$, where $\omega_k = u_k^T(\Gamma - \Psi)$, $k = 1, ..., M_1 < M$.
In a texture description, an image is viewed as a homogeneous 2D discrete random field, which by means of a Wold decomposition, is expressed as the sum of three orthogonal components. These components correspond to periodicity, directionality and randomness.
In creating a shape description, first a silhouette is extracted and a number of feature points on this are chosen (such as corners and high-curvature points). This feature points are then used as nodes in building a finite element model of the shape. Solving the following eigenvalue problem

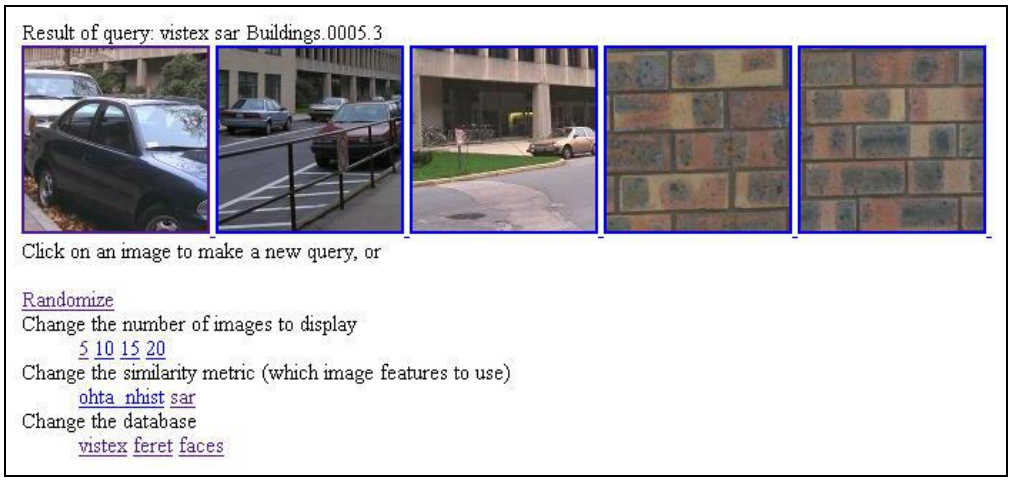

Figure 19: Photobook.

$K\phi_i = \omega_i^2 M\phi_i$, where M and K are the mass and stiffness matrices, respectively, the modes of the model are computed. These are the eigenvectors, ϕ_i, which are next used for determining a feature point correspondence between this new shape and some average shape.

Querying To perform a query, the user selects some images from the grid of still images displayed and/or enters an annotation filter. From the images displayed, the user can select another query images and reiterate the search.

Matching The distance between two eigenimage representations, Ω_1 and Ω_2, is $\epsilon_{ij}^2 = \|\Omega_i - \Omega_j\|^2$. Two shapes are compared by calculating the amount of strain energy needed to deform one shape to match the other.

Indexing Prior to any database search, a few prototypes that span the image category are selected. For any image in the database, its distance to the average of the prototypes is computed and stored for future database search. At query time, the distance of the query image to the average is computed and the database is reordered according to this.

Result presentation Images in the database are sorted by similarity with the query images and presented to the user page by page.

Applications The face recognition technology of Photobook has been used by Viisage Technology in a FaceID package, which is used in several US police departments.

26 Picasso

Developer Visual Information Processing Lab, University of Florence, Italy.

URL http://viplab.dsi.unifi.it/PICASSO. A web demo of the system can be evaluated at http://vesna.dsi.unifi.it/~picasso/picasso/.

References [BMPT97], [PS99].

Features The Picasso system supports queries based on shape, color regions and their spatial relationships. The system exploits a pyramidal color segmentation of the database images, each level of the pyramid corresponding to a resolution level of the segmentation. At the lowest level, each pixel in the image is identified with one region, while at the highest level of the pyramid the entire image is represented by a single region. Image regions at level n are determined by iteratively merging adjacent regions from level $n-1$, minimizing the sum of all region energies. The energy E_R of a region R is given by $E_R = \alpha 1/A(R) + \beta D(R) + \gamma \sum_{R_i \in N(R)} 1/D(R \cup R_i)$, where $A(R)$ is the region's area, $D(R)$ is a measure of the uniformity in color of the pixels belonging to the region R, $N(R)$ denotes the set of regions adjacent to R and α, β and γ are positive real weights. This way, at the end of the segmentation process, each image is associated with N segmented images

I_n, $n = 1 \ldots N$, which are represented using a multi-layered graph G. The nodes of G are the color regions of the N segmented images. There are two types of connections between these nodes: *intralevel* links between the nodes corresponding to adjacent regions in a segmented image I_n and *interlevel* links between a node correspondig to a region R_n^k of I_n and the nodes corresponding to regions of I_{n-1} that merged to form R_n^k. The description of a color region (graph node) include: a binary 187-dimensional color vector, the position of the region centroid, the region area (the fraction of region pixels in the total number of image pixels) and the region shape (the region is approximated by its best fit ellipse and the features used are the elongation and the major axis orientation). One property of the color vectors that helps in indexing and matching is that the color vector of a region R is computed as the union of the color vectors of the regions that merged to form R.

In order to allow queries by shape, the objects of interest in each image are bounded in the database population step with their minimum enclosing rectangle (MER). Edge images (128×128 in size) of these rectangular areas are extracted through a Canny edge detection. Also, spatial relationships between objects' MER are represented using 2D strings [CSY87]. The query contours drawn by the user are modeled by B-spline functions.

Querying The system interface allows the user to select the type of search: by color regions or by shape. In the first case, regions can be indicated in two ways: either by drawing contours, which are then filled with colors chosen from a palette, or by loading a database image and tracing the contour of some relevant regions in this image. The user can also adjust the relevance of the region attributes (position, area, elongation and orientation). A query by shape is based on user drawn sketches, while for a query by texture patterns, the user has to pick an image from the database.

Matching In a query by color regions, after a fast selection of the database images that contain all the colors of the query (see *Indexing*), the pyramidal structure of each candidate image is analyzed from top to bottom to find the best matching region for each query region. The matching score between a query region and a candidate image region is given by a weighted sum of distances between the computed region attributes (color, region centroid's position, area and shape). The similarity score between the query image and a candidate image is obtained by summing all the scores of the matched query regions.

In a shape based query, first images are filtered according to the spatial relationships and positions of the delimited MERs, based on the 2-D string representation. 1D elastic matching is applied to the images that have passed this filtering step. If the sketch contains k contours, the systems warps each contour over the candidate image's shape located in the same relative position as the query contour. The similarity score between the deformed contour and the image object takes into account both the match M between the deformed contour and the edge image and the amount of elastic energy E used to warp the query contour. A gradient descent technique is used in minimizing $E - M$.

Indexing In the color region queries, an expensive matching of the query against all database images is avoided by exploiting the color vectors. Because the color vector of the node in the highest level of the pyramidal color region representation includes colors of all the regions in the pyramid, a color index file is constructed. This contains the color vectors of the highest node of the pyramidal structure of each image and is used in the filtering step of the matching process, when only the images that contain all the colors of the query are selected for further inspection.

Result presentation The query results are presented to the user in decreasing similarity order.

Relevance feedback A query by sketch can be refined, for example by choosing one of the retrieved images and iterating a query by grey level intensity surface.

Applications The system is used for the electronic cataloging of paintings and sculptures of museums and galleries in the area of central Italy.

27 PicHunter

Developer NEC Research Institute, Princeton, NJ, USA.

References [CMM+00].

Features The current version of the systems uses color histogram and color spatial distribution along with hidden textual annotations. Besides a 64-bin HSV histogram, two other vectors – a 256-length HSV color autocorrellogram (CORR) and a 128-length RGB color-coherence vector (CCV) – are describing the color content of an image. The first 64 components of CORR represent the number of pixels of a particular color from the 64-quantized HSV color space having neighbors of the same color at distance 1. The rest of the vector is defined in the same way for distances 3, 5 and 7. CCV is made of 64 coherence pairs, each pair giving the number of coherent, and incoherent pixels of a particular discretized color in the RGB space. In order to classify pixels in one of the two categories, the image is first blurred slightly by replacing each pixel value with the average value of the pixels in its 3×3 neighborhood. Then, pixels are grouped in connected components. A pixel will be classified as coherent if the connected component it belongs to is larger than a fixed threshold. Keywords, selected from a set of 138 words, are associated with each image in the database population step and are represented as a boolean vector.

Querying Retrieval is done with query by example.

Matching The distance between individual features (color vectors or annotation lists represented as binary vectors) is the L_1 distance. These distances are scaled and combined in a global distance. The scaling factors are computed by maximizing the probability of a training set.

Relevance feedback PicHunter implements a probabilistic relevance feedback mechanism, which tries to predict the target image the user wants based on his actions (the images he selects as similar to the target in each iteration of a query session). A vector is used for retaining each image's probability of being the target. This vector is updated at each iteration of the relevance feedback, based on the history of the session (images displayed by the system and user's actions in previous iterations). The updating formula is based on Bayes' rule. If the n database images are noted T_j, $j = 1, \ldots, n$, and the history of the session through iteration t is denoted $H_t = \{D_1, A_1, D_2, A_2, \ldots, D_t, A_t\}$, with D_j and A_j being the images displayed by the system and, respectively, the action taken by the user at the iteration j, then the iterative update of the probability estimate of an image T_i being the target, given the history H_t, is:

$$P(T = T_i|H_t) = P(T = T_i|D_t, A_t, H_{t-1}) = \frac{P(A_t|T = T_i, D_t, H_{t-1})P(T = T_i|H_{t-1})}{\sum_{j=1}^{n} P(A_t|T = T_j, D_t, H_{t-1})P(T = T_j|H_{t-1})}.$$

In computing the probability of a user to take a certain action A_t given the history so far and the fact that the target is indeed T_i, namely $P(A_t|T = T_i, D_t, H_{t-1})$, a few models were tested. One approach is to estimate the probability of the user to pick an image X_a from $X_1, \ldots, X_{n_t}$ by

$$p_{softmin}(A = a|X_1, \ldots, X_{n_t}, T) = \frac{exp(-d(X_a, T)/\sigma)}{\sum_{i=1}^{n_t} exp(-d(X_i, T)/\sigma)},$$

and, in the case of choosing any number of images, to assume that each image is selected independently acording to a $p_{softmin}$.

Result presentation While older versions were displaying the n_t images with the highest probability of being the target, in the newer version the images selected for display are determined by minimizing the expected number of future iterations estimated by entropy.

Applications The system was tested with a database gathering images from 45 CD's of Corel stock photographs.

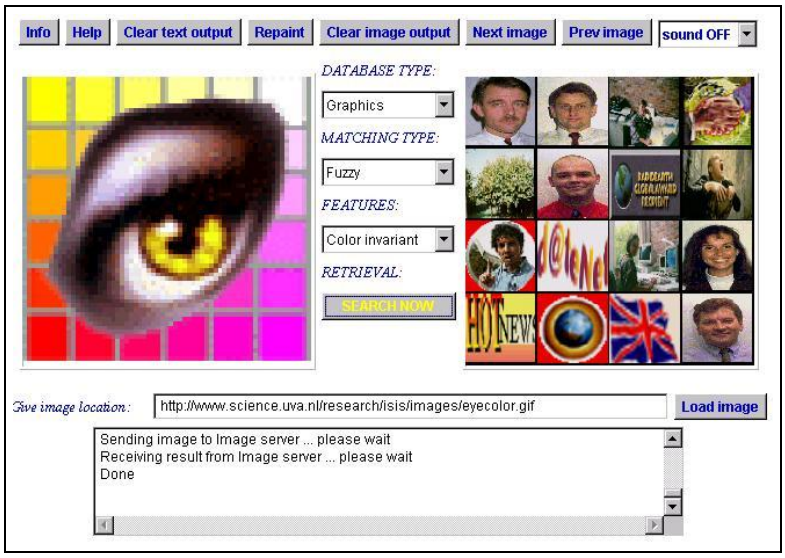

Figure 20: PicToSeek.

28 PicToSeek

Developer Department of Computer Science, University of Amsterdam, The Netherlands.

URL http://www.science.uva.nl/research/isis/pictoseek/. A demo of the system is available at http://zomax.wins.uva.nl:5345/ret_user/.

References [GS00].

Features Color and shape invariants are defined to be used as features in content based queries independent of camera viewpoint, geometry of the objects and illumination conditions. Under the assumption of a dichromatic reflection model, the effect of various imaging conditions on the different color features is analyzed. (In this model, reflection is used to determine how color clusters are formed in the RGB color space by pixels coming from the same uniformly colored surface of the object.) Assuming a white illumination and a matte object surface, a basic set of irreducible color invariants is formed by $\{R/G, B/R, G/B\}$. Also, it is proven that the normalized color model rgb and a newly defined model $c_4c_5c_6$, with $c_4 = (R-G)/(R+B)$, $c_5 = (R-B)/(R+B)$ and $c_6 = (G-B)/(G+B)$, are independent of the viewpoint, illumination direction, illumination intensity and surface orientation of matte objects. The set of color invariants, independent of viewpoint, used in the case of a shiny surfaced object, are $l_4 = |R-G|/(|R-G|+|B-R|+|G-B|)$, $l_5 = |R-B|/(|R-G|+|B-R|+|G-B|)$, $l_6 = |G-B|/(|R-G|+|B-R|+|G-B|)$. Based on the above color invariants, normalized color histograms are computed, H_A defined on the c_4, c_5 and c_6 axes and H_B on l_4, l_5, and l_6 axes. Color invariant gradients are defined using a multi-band approach and a histogram H_C of $l_4l_5l_6$ color edges is computed. That is because the color gradient for $l_4l_5l_6$ is the one that measures the presence of material edges (not shadow or highlight edges) in an image.

The shape invariants used are the angle between two color edges (invariant under rigid motion plus scaling) and the cross-ratio (projective invariant) between four color edges originating in the same point. A 1D histogram H_D is constructed on the discretized angle values, and a similar one H_E on the cross ratio values. Color and shape invariants are also combined in a 4D histogram H_F such that $H_F(i,j,k,l)$ counts the number of color edge maxima points of (i,j,k) color and generating an angle l.

Querying The query image can be selected from the database collected by the system's Web crawlers or brought in by the user by giving a URL address, see figure 20. Before submitting the query, the user has to select the desired invariance, which indicates to the system how to

choose the invariants in computing the similarity. Queries by user-specified feature values are also allowed.

Matching The similarity measure between a query histogram H_j^Q and a database image histogram H_j^I is the normalized intersection $d(H_j^Q, H_j^I) = (\sum_{k=1}^{n} min\{H_j^Q(k), H_j^I(k)\})/(\sum_{k=1}^{n} H_j^Q(k))$, with $j \in \{A, B, C, D, E, F\}$ and n is the number of nonzero bins in H_j^Q.

Result presentation The retrieved images are shown, without explicit order.

29 QBIC (Query By Image Content)

Developer IBM Almaden Research Center, San Jose, CA.

URL http://wwwqbic.almaden.ibm.com/. A demo version of the system is available at site http://wwwqbic.almaden.ibm.com/cgi-bin/stamps-demo.

References [NBE+93].

Features Color features computed are: the 3D average color vector of an object or the whole image in RGB, YIQ, Lab, and Munsell color space and a 256-dimensional RGB color histogram. If x is an n-dimensional color histogram and $C = [c_1 c_2 \dots c_n]$ is a $3 \times n$ matrix whose columns represent the RGB values of the n quantized colors, the average color vector x_{avg} is $C\ x$. The texture features used in QBIC are modified versions of the coarseness, contrast, and directionality features proposed by Tamura [TMY78].

The shape features consist of shape area, circularity, eccentricity, major axis orientation and a set of algebraic moment invariants. The major axis orientation and the eccentricity are computed from the second order covariance matrix of the boundary pixels: the major axis orientation as the direction of the largest eigenvector and eccentricity as the ratio of the smallest eigenvalue to the largest one. For the database images, these shape features are extracted for all the object contours, semiautomatically computed in the database population step. In this process, the user enters an approximate object outline, which is automatically aligned with the nearby image edges, using the active contours technique. In this object identification step, the user can also associate text to the outlined objects.

The 18 algebraic moment invariants are the eigenvalues of the matrices $M_{[2,2]}$, $M_{[2,3]} \times M_{[3,2]}$, $M_{[3,3]}$, $M_{[3,4]} \times M_{[4,3]}$, $M_{[4,4]}$, $M_{[4,5]} \times M_{[5,4]}$, where the elements of $M_{[i,j]}$ are scaled factors of the central moments.

QBIC also implemented a method of retrieving images based on a rough user sketch. For this purpose, images in the database are represented by a reduced binary map of edge points. This is obtained as follows: first, the color image is converted to a single band luminance; using a Canny edge detector, the binary edge image is computed and is next reduced to size 64×64. Finally this reduced image is thinned.

Querying QBIC allows queries based on example images, user-constructed sketches or/and selected color and texture patterns. In the last case, the user chooses colors or textures from a sampler. The percentage of a desired color in an image is adjusted by moving sliders.

Matching For the average color, the distance between a query object and database object is a weighted Euclidean distance, where the weights are the inverse standard deviation for each component over the samples in the database. In matching two color histograms, two distance measures are used: one low-dimensional, easy to compute (the average color distance) and one much more computationally expensive (the quadratic histogram distance). The first one (which is computed for all the images in the database) acts as a filter, limiting the expensive matching computation to the small set of images retrieved by the first matching. The average color distance is $d^2_{avg}(x, y) = (x_{avg} - y_{avg})^t (x_{avg} - y_{avg})$. The histogram quadratic distance is given by $d^2_{hist}(x, y) = (x-y)^t A (x-y)$, where the symmetric color similarity matrix A is given by $a_{ij} = 1 - d_{ij}/d_{max}$, with d_{ij} being the L_2 distance between the colors i and j in the RGB space and $d_{max} = max_{i,j} d_{ij}$. The texture distance is a weighted Euclidean distance, with the weighting factors being the inverse variances for each of the three texture components over the entire database. Two shapes are matched also by a similar weighted Euclidean distance between shape feature vectors.

Figure 21: QBIC.

In a query by sketch, after reducing the binary sketch image drawn by the user to size 64×64, a correlation based matching is performed. This is done by partitioning the user sketch into 8×8 blocks of 8×8 pixels and find the maximum correlation of each block of the sketch within a search area of 16×16 pixels in the image database (this is done by shifting the 8×8 block in the search area). This local correlation score is computed on the pixel level using logical operations. The matching score of a database image is the sum of the correlation scores of all local blocks.

Indexing QBIC was one of the first systems that applied multidimensional indexing to enhance the speed performance of the system. The average color and the texture features (both 3D vectors) are indexed using R^*-trees. The 18-dimensional moment-based shape feature vector is first reduced using the KL transform and then indexed by using R^*-trees.

Result presentation The best matches are presented in decreasing similarity order with (optionally) the matching score aside.

Relevance feedback Any retrieved image can be used as a seed for a subsequent query by example.

Applications At http://wwwqbic.almaden.ibm.com/tmdemo/ is a demonstration of the QBIC system as trademark server.

30 SQUID (Shape Queries Using Image Databases)

Developer Centre for Vision, Speech, and Signal Processing, University of Surrey, UK.

URL http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html. A demo is available at http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/squid.html.

References [MAK96].

Features The boundary contour extracted from an image is represented by three global shape parameters (eccentricity, circularity and aspect ratio of the Curvature Scale Space image) and a descriptor obtained from the CSS image. The CSS image of a closed planar curve, Γ, is computed by repeatedly convolving its natural parameterization, $\{(x(u), y(u)) \mid u \in [0,1]\}$, with a Gaussian function of increasing standard deviation σ and representing the curvature zero-crossing points (inflection points) of the resulting curve, $\Gamma_\sigma = \{(x(u) * g(u,\sigma)\,,\, y(u) * g(u,\sigma)) \mid u \in [0,1]\}$, in the (u,σ) plane. The shape descriptor is made of the maxima of the CSS image contours.

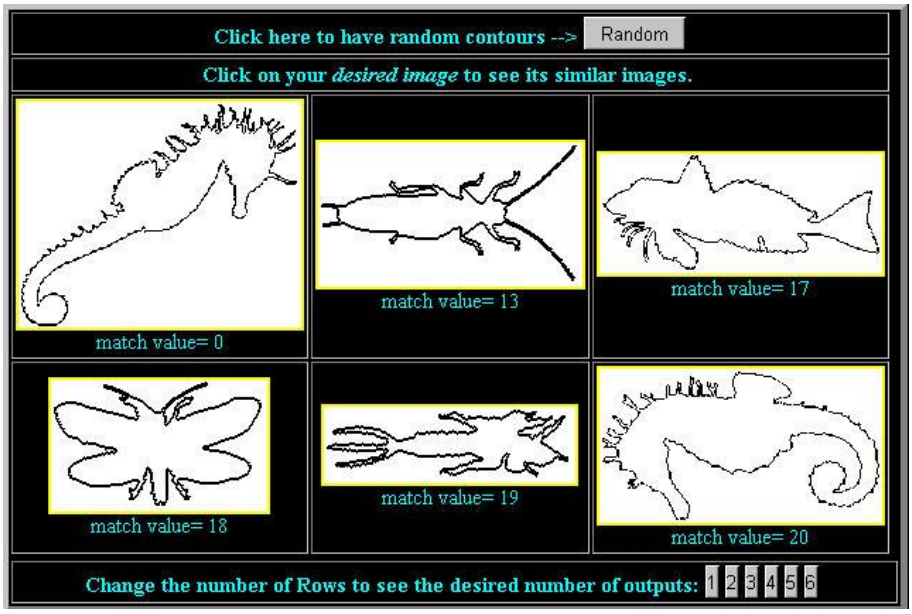

Figure 22: SQUID.

Querying The user selects one of the displayed object boundaries in order to generate a query for similar contours (see figure 22).

Matching First, from the database are chosen those models whose global parameters are sufficiently close to the query global parameters. To these images a CSS matching algorithm is applied. This consists of first finding a circular shift of one of the two sets of maxima so that they best match the other set and then computing the matching score as the sum of the Euclidean distances between the matched pairs plus the vertical coordinates of the unmatched maxima.

Result presentation The system displays the first n matched images in decreasing similarity order. Under each retrieved image, an integer match value is shown.

Applications The system database consists of 1100 images of marine creatures on a uniform background.

31 Surfimage

Developer INRIA, Rocquencourt, France.

URL `http://www-rocq.inria.fr/imedia/index_UK.html`. A demo of the system is available at `http://www-rocq.inria.fr/cgi-bin/imedia/surfimage.cgi`.

References [NMMB98].

Features Surfimage gives the possibility of combining various low-level and high-level features. The low-level features are RGB color histogram, edge-orientation histogram computed after applying a Canny edge detector, texture signature derived from the gray-level cooccurrence matrix, Fourier transform, and Wavelet transform. High-level features include eigenimages (see Photobook), deformable intensity surfaces, and intensity surface curvature histogram. The latter is a normalized histogram of

$$S_I(p) = \frac{1}{2} - \frac{1}{\pi} \arctan \frac{k_1(p) + k_2(p)}{k_1(p) - k_2(p)},$$

at each point $p = (x, y, I(x, y))$ over the entire image.

Querying The user selects image features and appropriate similarity measures from the lists displayed and specifies weights to define the importance of each feature in the retrieval. He also can

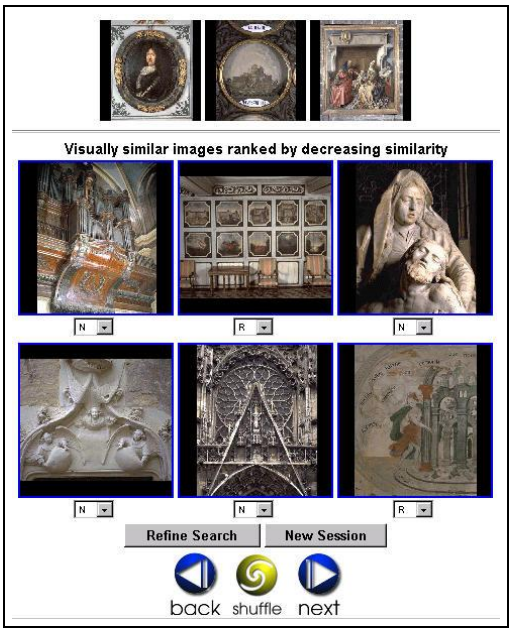

Figure 23: SurfImage.

select a database (among the databases available are the MIT face database used in Photobook, the MIT Vistex database of textures, a paintings database, the BTphoto database of city/country scenes and a combined database).

Matching Various similarity metrics are implemented in Surfimage. Among them are standard metrics, such as the L_p distances (most often used are L_1 and L_2), the cosine metric $d_{cos}(x, y) = 1 - x \cdot y/\|x\|\|y\|$, or statistical metrics such as the Hellinger metric

$$d_H(x, y) = 1 - \sum_{i=1}^{n} \sqrt{\frac{x_i\, y_i}{|x||y|}},$$

where $\|x\|$ is the L_2 norm and $|x|$ is the L_1 norm of the vector x. For histograms the intersection metric is used.

In the case of choosing more than one feature for the query, the corresponding similarity measures are combined in a final weighted measure. Before this, some of the distance measures are normalized so that they have values in $[0, 1]$. If μ_i and σ_i are the mean and standard deviation of the distance d for the feature i, this distance is normalized to: $d_n(x^i, y^i) = [d(x^i, y^i) - (\mu_i - 3\sigma_i)]/6\sigma_i$.

Relevance feedback Surfimage has a relevance feedback mechanism, which takes into consideration both relevant and nonrelevant examples from the user, in a statistical approach that estimates the distribution of relevant images.

32 SYNAPSE (SYNtactic APpearance Search Engine)

Developer Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA.

URL `http://hobart.cs.umass.edu/~mmedia/`. A demo of the system is available at the address `http://cowarie.cs.umass.edu/~demo/Demo.html`.

References [MR97].

Features Queries are made on the visual appearance of the objects in the image, defined as the shape of the intensity surface and represented using the outputs of a set of Gaussian derivative filters. More precisely, each image is filtered at uniformly sampled points with Gaussian derivative

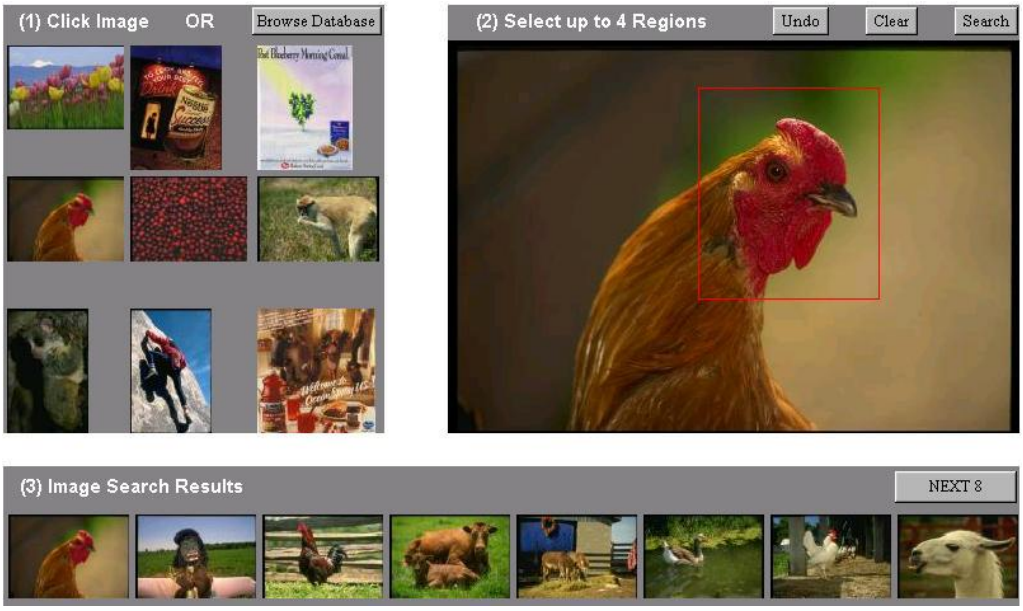

Figure 24: Synapse query with the drawn region of interest.

up to the order 2. Thus, for each sampled point $p = (x, y)$ of intensity $I(p)$, the so-called 2-jet at this point is given by the vector $J^2[I](p, \sigma) = \{I_\sigma(p), I_{x,\sigma}(p), I_{y,\sigma}(p), I_{x^2,\sigma}(p), I_{xy,\sigma}(p), I_{y^2,\sigma}(p)\}$, where $I_{x^i y^i,\sigma}(p)$ is the image convoluted with the (i, j)-th partial derivative of the Gaussian function of scale σ: $I_{x^i y^i,\sigma}(p) = I(p) * \sigma^{i+j} \partial^{i+j} G(p, \sigma)/\partial x^i \partial y^i$, $i, j \in \{0, 1, 2\}$. From this vector, four differential invariants are computed: $d_1^\sigma = I_x^2 + I_y^2$, $d_2^\sigma = I_{x^2} + I_{y^2}$, $d_3^\sigma = I_{x^2} I_x^2 + 2 I_{xy} I_x I_y + I_{y^2} I_y^2$, $d_4^\sigma = I_{x^2}^2 + 2 I_{xy}^2 + I_{y^2}^2$. Computing these four invariants at three different scales, $\sigma_1, \sigma_2, \sigma_3$, a multi-scale invariant vector D is generated for each sampled point p. Along with each multi-scale invariant vector D, the coordinates (x, y) of the corresponding sampled point and the number i of the image are also stored.

Querying Querying is done by example (see figure 24 where a region has been marked).

Matching Matching is done in two phases. First, for every invariant vector D_n of the query image, a number of invariant vectors stored in the database are found so that they lie in the range $[D_n - t, D_n + t]$. This is done by finding for each component $D_n(j)$ a list of invariant vectors, whose j-component lies in the interval $[D_n(j) - t(j), D_n(j) + t(j)]$ and intersecting the lists.

In the second phase, a spatial check is performed on the list of candidate images. For each sampled point q_m of the query image, the number S_m of query points whose Euclidean distance to the q_m is within a threshold factor of the Euclidean distance of their corresponding points in the candidate image. The score of the candidate image is given by $\max_m S_m$.

Indexing An index is created on each of the 12 coordinates of the invariant vector D. Instead of storing each invariant vector in one record of the form (i, x, y, D), 12 smaller tables (inverted file lists), each with 4 columns $(D(p), i, x, y)$, are generated. Each inverted file list is stored in a binary tree sorted on the first column.

Result presentation Images are showed in decreasing similarity score.

Relevance feedback The search can be refined by marking the relevant images found among the results of a query and reiterate the query.

33 TODAI (Typographic Ornament Database And Identification)

Developer Computer Science Department, EPFL, Switzerland.

URL http://www.unil.ch/BCU/docs/collecti/res_prec/en/wtres/todai_doc.html.

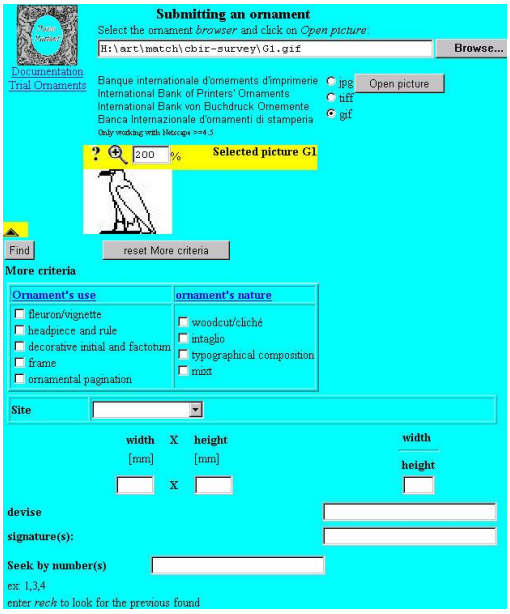

Figure 25: TODAI: query specification.

References [BBM96].

Features The image is first decomposed into six orientation images. Each orientation image is result of applying a filter that passes edges with a specified orientation. Thus, an orientation image contains information about the presence in the original image of linear structure in the direction of the pass orientation. The projection of the orientation image in the direction of the pass orientation is called an orientation radiogram. Six fixed directions are chosen to make radiograms. Each radiogram is represented by a feature vector of the first ten Fourier coefficients, excluding the discrete cosine coefficients.

Querying The user can specify the location of a query image. Additional criteria such as the size, use and nature of the ornament can be indicated, see figure 25.

Matching The similarity between two images is the sum of the Euclidean distances of the six feature vectors.

Result presentation The best match is shown, the user can browse through the list of next matches.

Applications The system is used in Passe-Partout, the International Bank of Printers' Ornaments (`http://www.unil.ch/BCU/docs/collecti/res_prec/fr/todai_intro.html`), which stores 735 images.

34 VIR Image Engine

The VIR Image Engine is an extensible framework for building content based image retrieval systems.

Developer Virage Inc.

URL `http://www.virage.com/products/vir-irw.html`.

References [BFG+96].

Features A basic concept is that of a *primitive*, which denotes a feature's type, computation and matching distance. Five abstract data types are defined: global values and histograms, local values and histograms, and graphs. The VIR Image Engine provides a set of general primitives, such as global color, local color, texture and shapes. Appart from these, various domain specific primitives can be created when developing an application. When defining such a primitive, the developer supplies a function for computing the primitive's feature data from the raw image.

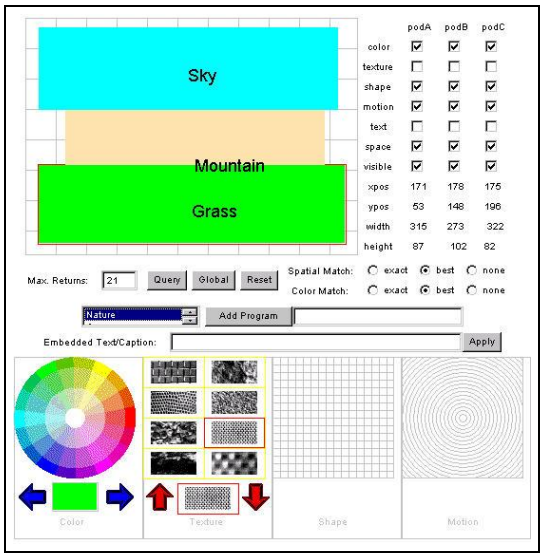

Figure 26: VisualSEEk.

Querying and Result presentation The VIR Image Engine provides a set of GUI tools necessary for the development of a user interface. These include facilities for image insertion, image query, weight adjustment for re-query, inclusion of keywords, and support for several popular image file formats. Another available component, the query canvas, allows queries-by-sketch; it consists of a bitmap editor where the user can sketch a picture with drawing tools and color it using the colors from a palette. Also, the user can bring onto the canvas an image from an existing collection and modify it using the same drawing tools. Queries can be performed on various user-defined combinations of primitives.

Matching When defining a new primitive, a function for computing the similarity between two sets of feature data previously extracted must also be supplied by the developer. When comparing two images, for each primitive in the current query combination, a similarity score is computed using the distance function defined within the primitive. These individual scores are combined in an overall score using a set of weights in a way characteristic to the application. This score is then stored in a *score structure*, which contains also the individual similarity scores for each primitive. This allows a quick recomputation of the overall score for a new set of weights.

Relevance feedback A sort of relevance feedback is obtained by searching for matches to a fixed query image for different sets of weights.

Indexing The storage of feature data for all primitives is the responsibility of the application developer.

Applications The system was integrated into databases from Sybase, Object Design, and Objectivity, and added as a component to the Oracle DBMS. Two applications of Virage technology are the AltaVista Photofinder and Illustra's Visual Intelligence system.

35 VisualSEEk

Developer Image and Advanced Television Lab, Columbia University, NY.

URL http://www.ctr.columbia.edu/VisualSEEk/.

References [SC97].

Features In the database population step, each image is automatically decomposed into regions of equally dominant colors. For each region, feature properties and spatial properties are retained for the subsequent queries. A query consists of finding the images that contain the most similar arrangements of similar regions.

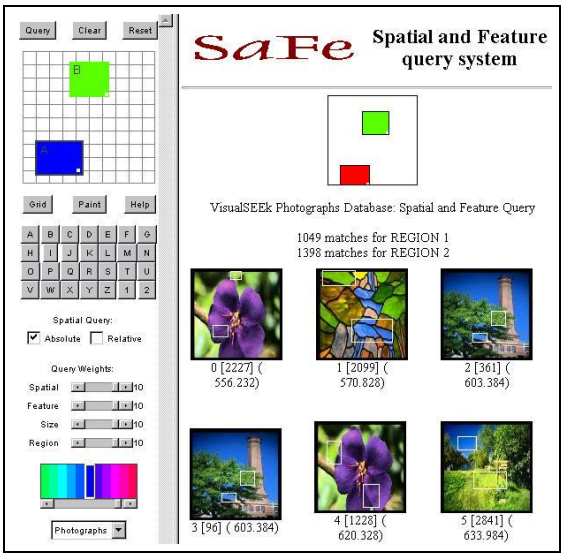

Figure 27: SaFe.

The color region extraction uses the back-projection technique. The first step is the selection of a color set. This is a 166-dimensional binary vector c, which defines a selection of 166 colors in the HSV color space. From a given image I, another image B is generated by $B[x,y] = \max_j a[k,j]c[j]$, where $k \in \{0,...,165\}$ is the index of the color of the pixel (x,y) in the image I and $a[k,j]$ is the similarity between two colors (with the indices k and j) in the HSV space. Next the image B is filtered and spatially localized color regions are extracted.
Along with the color set used for back-projection, the region centroid, area (defined as the number of pixels in the region) and the width and height of the minimum bounding rectangle are also stored.

Querying To start a query, the user sketches a number of regions, positions and dimensions them on the grid (see figure 26) and selects a color for each region. Also, the user can indicate boundaries for location and size and/or spatial relationships between regions. After the system returns the thumbnail images of the best matches, the user is allowed to search by example using the returned images.

Matching To find the matches of a query image with a single region, queries on color set, region absolute location, area and spatial extent are first done independently. The color set similarity is computed by $d(c_q, c_t) = (c_q - c_t)^t A(c_q - c_t)$, where c_q, c_t are two color sets and $A = (a[i,j])$ is the color similarity matrix. If the user has defined spatial boundaries for the query region, then its distance to a target region is 0 if the target region centroid falls inside this boundaries, and is given by the Euclidean distance between the centroids otherwise. The distance in area between two regions is the absolute value of the difference, while the distance between the minimum bounding rectangles, (w_q, h_q) and (w_t, h_t), of two regions is the L_2 metric. The results of these queries are intersected and from the obtained candidate set, the best matching images are taken by minimizing a total distance given by the weighted sum of the four distances mentioned.
If the query image consists of a number of regions, in absolute or relative location, then for each region positioned in absolute location, a query like that described above is made, and for regions positioned by relative location individual queries on all attributes except location are performed. For the intersection of all this query results, the relative spatial relations specified by the user are evaluated using 2D string representation [CSY87].

Indexing The color set distance can be written as $d(c_q, c_t) = \mu_q + \mu_t - 2c_q^t r_t$, where $\mu_q = c_q^t A c_q$, $\mu_t = c_t^t A c_t$ and $r_t = A c_t$. For each target region, μ_t and $r_t[m]$, $m \in \{0,...,165\}$, are indexed individually. The centroids of the image regions are indexed using a quadtree. For the indexing

Figure 28: VP.

of the minimum bounding rectangles, R-tree are used.

Result presentation The results of a query are displayed in decreasing similarity order. Under each retrieved image, the distance to the query image is indicated.

Performance Another system, SaFe [Smi97], see figure 27, was developed in order to improve the performances of VisualSEEk. SaFe was suppose to extend the query capabilities to other features, such as texture, and to incorporate more powerful spatial indexing techniques. There is no indication that this has been done. The two systems seems to differ only in the user interface.

36 VP Image Retrieval System

Developer National Center for Science Information Systems, University of Tokyo, Japan.

URL http://www.rd.nacsis.ac.jp/.

References [KSH95].

Features The silhouettes extracted from a database image are first decomposed into convex parts. The decomposition process is recursively performed as follows: at each negative curvature point V_i, an erosional vector, directed to the interior of the shape contour and proportional to the global angle and length of the concavity at V_i, is computed. Based on this vector, the speed to erode away the current part of the contour from a negative curvature point V_i to a point V_j on the other side of the contour is defined. The shape contour will be split by a line segment V_iV_j of minimum erosion time.

Next, for each part of the decomposition a number of features are extracted: *attributes* of the convex part (horizontal/vertical position, scale, shape, skeleton length, curvature and slope) and *relations* between two adjacent parts (relative horizontal/vertical position, relative scale, position of the skeletons intersection and angle between the two skeletons). In order to represent these features, their value domain is quantized into a small number of levels (e.g., 3 for the length attribute, which correspond to small, medium, large), each feature being associated a fixed number of bits in a binary sequence. Two binary vectors are constructed. The first one is the concatenation of all attribute sequences of the parts in the decomposition (such a sequence is called primitive signature). The second one, retains the relations between the parts along with their attributes.

Querying Queries are formulated in three steps. First, a silhouette is drawn, this is next automatically decomposed into corresponding convex parts, and finally the user associates query weights to these parts. These weights specify the degree of relevance in the query process, from not important to very important, of the computed attributes and relations of the query parts. When

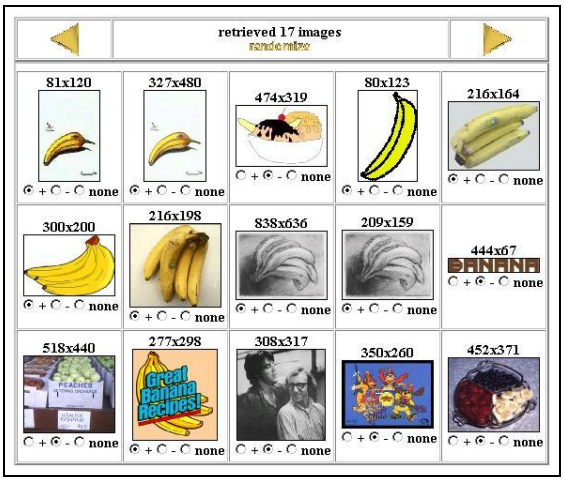

Figure 29: WebSEEk relevance feedback.

the user chooses to relax some of the attributes of particular parts, the system represents this by adding extra 1 bit values to the primitive signatures of these parts in the sequences coresponding to the selected attributes.

Matching The matching is done by shifting each query primitive signature over the concatenated primitive signatures of the target image and counting the matches C_a. Similar shifting and conjunction operations are applied to the second vector, storing the relations between parts, with the number of matched relations computed being C_r. The similarity score is given by $S = (C_a + C_r)/(N_a + N_r)$, where N_a is the total number of primitive signatures of the first query vector, and N_r is the total number of relations in the other query vector.

37 WebSEEk

Developer Image and Advanced Television Lab, Columbia University, NY.

URL http://www.ctr.columbia.edu/WebSEEk/.

References [Smi97]

Features WebSEEk makes text-based and color based queries through a catalogue of images and videos collected from the Web. Color is represented by means of a normalized 166-bin histogram in the HSV color space, see VisualSEEk.

Querying The user initiates a query by choosing a subject from the available catalogue or entering a topic. The results of the query may be used for a color query in the whole catalogue or for sorting the result list by decreasing color similarity to the selected item. Also, the user has the possibility of manually modifying an image/video color histogram before reiterating the search.

Matching The distance between a query color histogram h_q and a target histogram h_t is given by $d(h_q, h_t) = (h_q - h_t)^t A(h_q - h_t) \approx \mu_q + \mu_t - 2\sum_{k\ with\ h_q[k] \geq \tau} h_q[k] r_t[k]$, where $A = (a[i,j])$ is a color similarity matrix, $\mu_q = h_q^t A h_q$, $\mu_t = h_t^t A h_t$, $r_t = A h_t$ and τ defines a threshold for choosing the most significant colors in the approximated distance.

Indexing For all the database images, μ_t and $r_t[k]$, $k \in \{0, ..., 165\}$ are computed and indexed individually.

Result presentation Images retrieved are displayed page by page in a descending similarity order. A manipulation (union, intersection, subtraction) of the search result lists, in order to reiterate a query, is possible.

Relevance feedback The user has the possibility of selecting positive and negative examples from the result of a query in order to reformulate the query, see figure 29. If the set of relevant images

is denoted by I_r and the set of nonrelevant images is I_n then the new query histogram at the $(k+1)$th feedback iteration is computed by $h_q^{k+1} = \|\alpha h_q^k + \beta \sum_{i \in I_r} h_i - \gamma \sum_{j \in I_n} h_j\|$.

38 WebSeer

Developer Department of Computer Science, University of Chicago, Illinois, USA.

URL http://infolab.cs.uchicago.edu/webseer/.

References [SFA96].

Features The images collected from the Web are submitted to a number of color tests in order to separate photographs from drawings. Some simple tests measure the number of different colors in the image, the fraction of pixels which have one of the N most frequent colors for a given threshold N, the fraction of pixels with transition value (the largest L_1 distance between a pixel and one of its neighbors in the RGB space) greater than a threshold T, the fraction of pixels with saturation level (the difference between the maximum and the minimum value of the RGB color bands) greater than a threshold T, and the ratio of the image dimensions. A more elaborate test creates first an average color histogram for graphics, H_g, and one for photographs, H_p, using two large sets of images. Defining the correlation between two normalized RGB histograms, A and B, as $C(A,B) = \sum_{i=0}^{15} \sum_{j=0}^{15} \sum_{k=0}^{15} A_{i,j,k} B_{i,j,k}$, an image with a color histogram H_i gains a score to the test equal to $s = C(H_i, H_p)/(C(H_i, H_p) + C(H_i, H_g))$. Two similar tests are also made using the farthest neighbour histograms and the saturation histograms instead of the color histograms.

Images determined to be photographs are subjected to a face detector based on a neural network. Keywords are extracted from the image file name, captions, hyperlinks, alternate text and HTML titles.

Querying The user gives keywords describing the contents of the desired images, and optionally specifies some image characteristics such as dimensions, file size or whether he is looking for photographs or drawings. In the case the user is looking for people, he must indicate the number of faces as well as the size of the portrait.

Matching To classify an image as photograph or graphics, the color tests are combined using multiple decision trees constructed using a training set of hand-classified images. They are binary trees whose internal nodes contain the next test the image should be submitted to and a threshold to direct the search by comparing the test result with the threshold. In each leaf we find a probability estimate that the image is a photograph. Computing the mean of the results got from all decision trees and comparing this with a threshold, a decision is taken whether the image falls in one category or the other.

Result presentation The thumbnails of the resulting images are displayed by decreasing size of the face in the case of searching for portraits. Otherwise, there is no explicit order of the retrieved images. The user has access to the Web page where the image was collected from, through a page icon aside the displayed thumbnail.

39 WISE (Wavelet Image Search Engine)

Developer Department of Computer Science, Stanford Univerity.

URL http://www-db.stanford.edu/~wangz/project/imsearch/. A demo of the system is available at http://bergman.stanford.edu/~zwang/project/imsearch/WBIIS.html

References [WWFW97].

Features The system, also called WBIIS (Wavelet-Based Image Indexing and Searching), performs queries on color layout information encoded using Daubechies wavelet transforms. In a preprocessing step, the image is rescaled to 128×128 pixels and converted from RGB color space representation to another color space. This color space, defined by $C_1 = (R+G+B)/3$, $C_2 = (R+(max-B))/2$, $C_3 = (R+2*(max-G)+B)/4$, with max the maximum possible value for each RGB color component, is selected because of its similarity to the opponent color axes used by the human visual system. On each of the three color components, a 4-layer 2D fast wavelet transform using Daubechies wavelets is applied. From the resulting 128×128 matrices, denoted

Figure 30: WISE.

as W_{C_1}, W_{C_2}, and W_{C_3}, each 16×16 upper left corner is stored as part of the feature vector (the 8×8 upper left corner, representing the lowest frequency bands in the wavelet transforms, accounts for object configurations in the image, while the three surrounding 8×8 submatrices represent detail information). Also, the standard deviation, σ_{C_i}, of the 8×8 upper left corner of W_{C_i} is computed and stored as part of the feature vector.

Querying Querying is done by example, where the query image is either a database image or a hand drawn sketch. Also low resolution images or partial images (images that contain non-specified areas represented by black rectangles) can be query images to the system.

Matching The matching process is performed in two steps. In the first step, the standard deviations computed for the query image, $\sigma^q_{C_i}$, are compared with the corresponding values, $\sigma^i_{C_i}$, stored for each database image. If the acceptance criterion, $(\sigma^q_{C_1}\beta < \sigma^i_{C_1} < \sigma^q_{C_1}/\beta) \vee ((\sigma^q_{C_2}\beta < \sigma^i_{C_2} < \sigma^q_{C_2}/\beta) \wedge (\sigma^q_{C_3}\beta < \sigma^i_{C_3} < \sigma^q_{C_3}/\beta))$, with the threshold β usually set to 0.5, is not achieved, the distance between the two images is set to 1. In the second step, the images that passed the first matching phase are compared to the query using a weighted Euclidean distance between wavelet transform coefficients: $\sum_{m=1}^{2}\sum_{n=1}^{2} w_{mn} \sum_{i=1}^{3}(w_{C_i} d_E(W^i_{C_i,m,n}, W^q_{C_i,m,n}))$, where $W_{C_i,1,1}$ and $W_{C_i,1,2}, W_{C_i,2,1}, W_{C_i,2,2}$ denotes the 8×8 upper left corner and, respectively, the three surrounding 8×8 submatrices of W_{C_i}, and with d_E the Euclidean distance. By raising w_{C_2} or w_{C_3} the importance of color variation in the image is emphasized, while raising $w_{2,1}, w_{1,2}$ or $w_{2,2}$ the vertical, horizontal, or diagonal edge details in the image are emphasized.

Result presentation The first few matches are presented in decreasing similarity order.

Applications The system is used at Stanford University Library to assist teaching and research projects in liberal art departments.

4 Summary

The features used in the 39 systems discussed in this article are listed in table 1. The features are classified into the low level classes color, texture, and shape, and the higher level classes layout and face detection. The use of keywords is also listed. We have tried to group the low level features into meaningful categories. For example, various forms of choosing important colors are grouped

into 'dominant colors'. The 'eigen image' is listed under color, because it is a feature derived from the global image color values, rather than local patterns as in texture. The 'atomic texture features' are features such as contrast, anisotropy, density, randomness, directionality, softness, and uniformity, often variations of the Tamura features, and often derived from a cooccurrence matrix of pixel values. The 'elementary shape descriptors' are features such as centroid, area, major axis orientation, eccentricity, circularity, and features derived from algebraic moments. The feature 'layout' is the absolute or relative spatial position of the color, texture, or shape features. Taking color histograms per fixed subimage is a simple way of exploiting spatial layout; considering the relative position (constellation) of regions is more complex.

Of the 39 systems in the table, 31 use any kind of color features, 25 use texture, 23 use shape, 12 layout, and 4 use face detection. Perhaps the reason for this order of frequencies of usage is their complexity and, coupled to that, effectiveness. The simpler the features can be extracted, the easier it is to implement it in the system, and the easier it is to use that feature class in the right way. For example, color features are often found effective, because good color features are not very difficult to design and implement. However, shape features that are for example robust against noise and occlusion, are more involved. As a result, only overly simple features such as area are used, which is often little effective.

		ADL	AltaVista	Amore	BDLP	Blobworld	CANDID	C-bird	Chabot	CBVQ	DrawSearch	Excalibur	FIR	FOCUS	ImageFinder	ImageMiner	ImageRETRO	ImageRover	ImageScape	Jacob	LCPD	MARS	MetaSEEk	MIR	NETRA
	Keywords	⋆		⋆	⋆				⋆									⋆				⋆	⋆	⋆	
	Face Detection																		⋆					⋆	
	Layout					⋆		⋆						⋆		⋆		⋆	⋆			⋆		⋆	⋆
Shape	No details		⋆				⋆					⋆													
	Other			[1] ⋆				[4] ⋆									[6] ⋆								
	Elementary descriptors					⋆		⋆		⋆				⋆		⋆			⋆						
	Bounding box/ellipse															⋆									
	Curvature scale space																								
	Elastic models																								
	Fourier descriptors										⋆								⋆			⋆			⋆
Texture	No details		⋆				⋆					⋆			⋆										
	Other																								
	Projection histogram																			⋆					
	Edge-orientation histogram							⋆										⋆							
	Local binary patterns																		⋆		⋆				
	Random fields										⋆														
	Atomic texture features					⋆										⋆		⋆	⋆	⋆		⋆	⋆		
	Wavelet, Fourier transform	⋆								⋆			⋆									⋆			⋆
Color	No details		⋆				⋆					⋆			⋆										
	Other				[2] ⋆			[3] ⋆									[5] ⋆		[7] ⋆						
	Dominant colors			⋆				⋆								⋆			⋆						
	Region histogram					⋆																			⋆
	Fixed subimage histogram													⋆				⋆				⋆			
	Color coherence vectors																								
	Average color vectors										⋆						⋆								
	Correllation histogram																							⋆	
	Global histogram							⋆	⋆	⋆							⋆	⋆		⋆		⋆	⋆	⋆	
	Eigen image																								
	System	ADL	AltaVista	Amore	BDLP	Blobworld	CANDID	C-bird	Chabot	CBVQ	DrawSearch	Excalibur	FIR	FOCUS	ImageFinder	ImageMiner	ImageRETRO	ImageRover	ImageScape	Jacob	LCPD	MARS	MetaSEEk	MIR	NETRA

System	*Color*										*Texture*							*Shape*							*Layout*	*Face Detection*	*Keywords*
	Eigen image	Global histogram	Correlation histogram	Average color vectors	Color coherence vectors	Fixed subimage histogram	Region histogram	Dominant colors	Other	No details	Wavelet, Fourier transform	Atomic texture features	Random fields	Local binary patterns	Edge-orientation histogram	Projection histogram	Other	No details	Fourier descriptors	Elastic models	Curvature scale space	Bounding box/ellipse	Elementary descriptors	Other	No details			
Photobook	★												★							★							★	
Picasso							★													★		★	★			★		
PicHunter		★	★		★																							★
PicToSeek		★																						★[8]				
QBIC		★		★								★											★			★		★
SQUID																					★							
Surfimage	★	★									★	★			★		★[9]			★								
SYNAPSE																	★[10]											
TODAI																★												
VIR										★								★							★			★
VisualSEEk								★														★	★			★		
VP																								★[11]				
WebSEEk		★																										★
WebSeer		★							★[12]																		★	★
WISE											★																	

Table 1: Overview of systems and their search features.

1 region overlap
2 number of regions with that color
3 DCT of chromaticity
4 generalized Hough transform
5 color variation, greyness, average saturation, background proportion
6 number of regions, number of holes
7 Laplacian
8 angles between edges, and cross ratios of them
9 intensity surface curvature histogram
10 multiscale intensity surface derivatives
11 convex parts
12 farthest neighbor histogram

5 Conclusions

Most systems are products of research, and therefore emphasize one aspect of content-based retrieval. Sometimes this is the sketching capability in the user interface, sometimes it is the new indexing data structure, etc. Some systems exist in a research version and in a commercial or production version. The commercial version is usually less advanced, and shows more standard searching capabilities. For example, a research version of Amore exhibits sketching and more fancy result visualization than is shown in the Arthur application system.

A number of systems provide a user interface that allows more powerful query formulation than is useful in the demo system. For example, if a user can paint a cartoon, but the database contains only airplanes, the system will always retrieve images of airplanes. For such a limited database, no powerful sketching interface is needed. Also, because most workstations have a mouse, but easy sketching needs a pencil, such a painting tool is often of little use. Drawing polygonal shapes with a mouse works well, however.

Most systems use color and texture features, few systems use shape feature, and still less use layout features. The retrieval on color usually yield images with similar colors. Retrieval on texture does not always yield images that have clearly the same texture, unless the database contains many images with a dominant texture. Searching on shape gives often suprising results. Apparently the shape features used for matching are not the most effective ones.

Indexing data structures are often not used. Indeed, for small collections of images, an indexing data structure is not needed, and a linear search can be sufficiently fast. Contemporary computers can perform simple matching of hundreds of images in near real time.

It is difficult to evaluate how successful content-based image retrieval systems are, in terms of effectiveness, efficiency, and flexibility. Of course there are the notions of precision (the ratio of relevant images to the total number of images retrieved) and recall (the percentage of relevant images among all possible relevant images). Many articles about systems give figures about precision and recall. Most of them are good, but hard to verify. One reason is that many hyperlinks on the Web are not active anymore, a design flaw of the Web. However, there are also considerations intrinsic to retrieval systems.

If the database only contains fighting airplanes, and the user can only ask for images similar to a chosen fighting airplanes, the system will successfully return fighting airplanes. If the domain is so narrow, it may make more sense to look for dissimilar images than for similar images. On the other hand, if the database is very diverse and contains only a single image of a chicken, asking for images similar to that chicken will not result in other chicken images. The larger the collection of images, the more chance that it contains an image similar to the query image. The Web is a large enough test set, and free of charge, reason why some image retrieval systems exploit webcrawlers. Having a specific object in mind, looking for images with similar objects is a frustrating experience, however. Indeed, first you crawl, than you learn to walk.

It is widely recognized that most current content-based image retrieval systems work with low level features (color, texture, shape), and that next generation systems should operate at a higher semantic level. One way to achieve this is to let the system recognize objects and scenes. Although this is difficult in general, it should be feasible for specific domains. For example, the ImageMiner system recognizes landscapes, and body plans have been used to recognize animals and people [FF97]. Once separate entities in images are recognized, the way to semantic reasoning lies open.

References

[Bal81] D. H. Ballard. Generalized Hough transform to detect arbitrary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):111–122, 1981.

[BBC98] A. B. Benitez, M. Beigi, and S.-F. Chang. Using relevance feedback in content-based image metasearch. *IEEE Internet Computing*, 2(4):59–69, July/August 1998.

[BBM96] J. Bigun, S. K. Bhattacharjee, and S. Michel. Orientation radiograms for image retrieval. In *International Conference on Pattern Recognition, ICPR-96*, volume C, pages 346–350. IEEE Computer Society, 1996.

[BFG+96] J. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Gorowitz, R. Humphrey, R. Jain, and C. Shu. Virage image search engine: an open framework for image management. In *Proceedings of the SPIE, Storage and Retrieval for Image and Video Databases IV, San Jose, CA*, pages 76–87, February 1996.

[BKSS90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 322–331, 1990.

[BL99] Jean Marie Buijs and Michael Lew. Visual learning of simple semantics in imagescape. In Huijsmans and Smeulders [HS99], pages 131–138.

[BMPT97] A. Del Bimbo, M. Mugnaini, P. Pala, and F. Turco. Picasso: Visual querying by color perceptive regions. In *Proceedings of the 2nd International Conference on Visual Information Systems, San Diego, December '97*, pages 125–131, 1997.

[CA96] M. La Cascia and E. Ardizzone. Jacob: Just a content-based query system for video databases. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP96, May 7-10, '96, Atlanta, Georgia*, 1996.

[CMM+00] Ingemar J. Cox, Matthew L. Miller, Thomas P. Minka, Thomas Papathomas, and Peter N. Yianilos. The bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments. *IEEE Transactions on Image Processing (to appear)*, 2000.

[CO96] Chad Carson and Virginia E. Ogle. Storage and retrieval of feature data for a very large online image collection. *IEEE Computer Society Bulletin of the Technical Committee on Data Engineering*, 19(4):19–27, December 1996.

[CSY87] S.-K Chang, Q.Y. Shi, and C.Y. Yan. Iconic indexing by 2-d strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):413–428, May 1987.

[CTB+99] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In Huijsmans and Smeulders [HS99].

[DRD97] M. Das, E. M. Riseman, and B. Draper. Focus: Searching for multi-colored objects in a diverse image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition '97*, pages 756–761, June 1997.

[EG99] John P. Eakins and Margaret E. Graham. Content-based image retrieval, a report to the JISC technology application programme. Technical report, Institute for Image Data Research, University of Northumbria at Newcastle, UK, January 1999. http://www.unn.ac.uk/iidr/report.html.

[FF97] David A. Forsyth and Margaret M. Fleck. Body plans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 678–683, 1997.

[GR95] V. N. Gudivada and V. V. Raghavan. Content-based image retrieval systems. *IEEE Computer*, 28(9):18–31, September 1995.

[Gro94] W. I. Grosky. Multimedia information systems. *IEEE Multimedia*, 1(1):12–24, 1994.

[GS00] Theo Gevers and Arnold Smeulders. Pictoseek: Combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing*, 9(1):102–119, January 2000.

[HHSH93] Kyoji Hirata, Yoshinori Hara, Naoki Shibata, and Fusako Hirabayashi. Media-based navigation for hypermedia systems. In *Proceedings of the fifth ACM conference on Hypertext, November 14-18, '93, Seattle, WA, USA*, pages 159–173, 1993.

[HMO+97] Kyoji Hirata, Sougata Mukherjea, Yusaku Okamura, Wen-Syan Li, and Yoshinori Hara. Object-based navigation: An intuitive navigation style for content-oriented integration environment. In *Proceedings of the eighth ACM conference on Hypertext, '97, Southampton, UK*, pages 75–86, 1997. http://journals.ecs.soton.ac.uk/~lac/ht97/.

[HS99] D. P. Huijsmans and A. W. M. Smeulders, editors. *Visual Information and Information Systems, Proceedings of the Third International Conference VISUAL '99, Amsterdam, The Netherlands, June 1999*, Lecture Notes in Computer Science 1614. Springer, 1999.

[Jai96] R. Jain. Infoscopes: Multimedia information system. In B. Furht, editor, *Multimedia Systems and Techniques*, pages 217–253. Kluwer, 1996.

[KCH95] P. M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: the CANDID approach. In *SPIE Vol. 2420, Storage and Retrieval for Image and Video Databases III*, pages 238–248, 1995.

[KRA+97] J. Kreyss, M. Röper, P. Alshuth, Th. Hermes, and O. Herzog. Video retrieval by still image analysis with ImageMiner. In *Proceedings of IS&T/SPIE's Symposium on Electronic Imaging: Science & Technologie, 8-14 Feb. '97, San Jose, CA*, 1997.

[KS91] H. F. Korth and A. Silberschatz. *Database System Concepts*. McGraw-Hill, 1991.

[KSH95] Fumikazu Kanehara, Shin'ichi Satoh, and Takashi Hamada. A flexible image retrieval using explicit visual instruction. In *Proceedings of the Third International Conference on Document Analysis Recognition, Montreal, Canada, August '95*, pages 175–178, 1995.

[LHD96] Michael S. Lew, D. P. Huijsmans, and Dee Denteneer. Content based image retrieval: KLT, projections, or templates. In Smeulders and Jain [SJ96], pages 27–34.

[LLH97] Michael Lew, Kim Lempinen, and Nies Huijmans. Webcrawling using sketches. In *Proceedings of the 2nd International Conference on Visual Information Systems, San Diego, December '97*, pages 77–84, 1997.

[LZT99] Z.N. Li, O. R. Zaïane, and Z. Tauber. Illumination invariance and object model in content-based image and video retrieval. *Journal of Visual Communication and Image Representation*, 10(3):219–244, September 1999. http://www.cs.sfu.ca/cs/undergrad/CM/CMPT365/CBS/CBIRD.pdf.

[LZY98] Z. N. Li, O. R. Zaïane, and B. Yan. C-bird: Content-based image retrieval from image repositories using chromaticity and recognition kernel. Technical Report CMPT98-03, School of Computing Science, Simon Fraser University, Canada, February 1998. ftp://ftp.fas.sfu.ca/pub/cs/TR/1998/CMPT1998-03.ps.gz.

[Ma97] W. Y. Ma. *NETRA: A Toolbox for Navigating Large Image Databases*. PhD thesis, Dept. of Electrical and Computer Engineering, University of California at Santa Barbara, June 1997.

[MAK96] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In Smeulders and Jain [SJ96], pages 35–42.

[Man95] B. S. Manjunath. Image browsing in the alexandria digital library project. *D-Lib Magazine*, August 1995. http://www.dlib.org/dlib/august95/alexandria/08manjunath.html.

[MHH97] Sougata Mukherjea, Kyoji Hirata, and Yoshinori Hara. Towards a multimedia world wide web information retrieval engine. In *Sixth International WWW Conference, 7-11 April '97, Santa Clara, CA, USA*, 1997. http://decweb.ethz.ch/WWW6/Technical/Paper003/Paper3.html.

[MHH99] Sougata Mukherjea, Kyoji Hirata, and Yoshinori Hara. Amore: A world wide web image retrieval engine. *The WWW Journal*, 2(3):115–132, 1999. http://www.baltzer.nl/www/contents/1999/2-3.html.

[MM99] Wei-Ying Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[MR97] R. Manmatha and S. Ravela. A syntactic characterization of appearance and its application to image retrieval. In *Proceedings of the SPIE conference on Human Vision and Electronic Imaging II, Vol, 3016, San Jose, CA, Feb. '97*, 1997.

[NBE+93] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The qbic project: Quering images by content using color, texture, and shape. In *Poceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases, 2-3 February '93, San Jose, CA*, pages 173–187, 1993.

[NMMB98] Chahab Nastar, Matthias Mitschke, Christophe Meilhac, and Nozha Boujemaa. Surfimage: A flexible content-based image retrieval system. In *Proceedings of the ACM International Multimedia Conference, 12-16 September '98, Bristol, England*, pages 339–344, 1998. http://www-rocq.inria.fr/_nastar/MM98/.

[ORC+97] Michael Ortega, Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, and Thomas S. Huang. Supporting similarity queries in MARS. In *Proceedings of the 5th ACM International Multimedia Conference, Seattle, Washington, 8-14 Nov. '97*, pages 403–413, 1997.

[OS95] Virginia E. Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48, September 1995.

[PlaBC] Plato. *Meno*. 380 B.C. Perseus Encyclopedia, Tuft University, http://www.perseus.tufts.edu/Texts/chunk_TOC.grk.html.

[PPS96] Alex Pentland, Rosalind W. Picard, and Stanley Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, June 1996.

[PS99] P.Pala and S. Santini. Image retrieval by shape and texture. *Pattern Recognition*, 32(3):517–527, 1999.

[RHC99] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(1):1–23, March 1999.

[SC95] J. R. Smith and S.-F. Chang. Automated image retrieval using color and texture. Technical Report CU/CTR 408-95-14, CTR, Columbia University, July 1995.

[SC97] J. R. Smith and S.-F. Chang. Querying by color regions using the VisualSEEk content-based visual query system. In M. T. Maybury, editor, *Intelligent Multimedia Information Retrieval*. AAAI Press, 1997.

[SFA96] Michael J. Swain, Charles Frankel, and Vassilis Athitsos. Webseer: An image search engine for the world wide web. Technical Report TR-96-14, Deptartment of Computer Science, University of Chicago, July 1996.

[SJ96] A. W. M. Smeulders and R. Jain, editors. *Image Databases and Multi-Media Search, proceedings of the First International Workshop IDB-MMS'96, Amsterdam, The Netherlands*. Amsterdam University Press, August 1996.

[Smi97] J. R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis*. PhD thesis, Graduate School of Arts and Sciences, Columbia University, February 1997.

[SMM99] E. Di Sciascio, G. Mingolla, and M. Mongiello. Content-based image retrieval over the web using query by sketch and relevance feedback. In Huijsmans and Smeulders [HS99], pages 123–130.

[Sri95] Rohini Srihari. Automatic indexing and content-based retrieval of captioned images. *IEEE Computer*, 28(9):49–56, September 1995.

[STC97] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content-based image browser for the world wide web. In *Proceedings IEEE Workshop on Content-based Access of Image and Video Libraries, June '97*, 1997.

[SZR00] Rohini Srihari, Zhongfei Zhang, and Aibing Rao. Intelligent indexing and semantic retrieval of multimodal documents. *Information Retrieval*, 2(2):245–275, 2000. http://www.cse.buffalo.edu/~arao/Papers/ir.ps.gz.

[TCS97] Leonid Taycher, Marco La Cascia, and Stan Sclaroff. Image digestion and relevance feedback in the ImageRover WWW search engine. In *Proceedings of the 2nd International Conference on Visual Information Systems, San Diego, December '97*, pages 85–94, 1997.

[TMY78] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–473, 1978.

[TY84] Hideyuki Tamura and Naokazu Yokoya. Image database systems: A survey. *Pattern Recognition*, 17(1):29–43, 1984.

[Ven97] J. Vendrig. Filter image browsing: a study to image retrieval in large pictorial databases. Master's thesis, Dept. Computer Science, University of Amsterdam, The Netherlands, February 1997. http://carol.wins.uva.nl/~vendrig/thesis/.

[VH99] Remco C. Veltkamp and Michiel Hagedoorn. State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, Department of Computer Science, September 1999. http://ftp.cs.uu.nl/pub/RUU/CS/techreps/CS-1999/1999-27.ps.gz

In Michael Lew, editor, *Principles of Visual Information Retrieval*. Springer, 2000.

[Vol97] S. Volmer. Tracing images in large databases by comparison of wavelet fingerprints. In *Proceedings of the 2nd International Conference on Visual Information Systems, San Diego, December '97*, pages 163–172, 1997.

[VWS99] J. Vendrig, M. Worring, and A. W. M. Smeulders. Filter image browsing: Exploiting interaction in image retrieval. In Huijsmans and Smeulders [HS99], pages 147–154.

[WHS99] Z. Wang, L. Hill, and T. Smith. Alexandria digital library metadata creator with extensible markup language. In *Proceedings of the Third International Conference on Conceptions of Library and Information Science (CoLIS 3). Digital Libraries: Interdisciplinary Concepts, Challenges and Opportunities. Dubrovnik, Croatia*, 1999.

[WJ96] D. White and R. Jain. Similarity indexing with the SS-tree. In *Proceedings of the 12th International Conference on Data Engineering, New Orleans, LA*, 1996.

[WWFW97] James Ze Wang, Gio Wiederhold, Oscar Firschein, and Sha Xin Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. In *Proceedings of the Fourth Forum on Research and Technology Advances in Digital Libraries, Washington D.C., May '97*, pages 13–24, 1997. http://www-db.stanford.edu/~wangz/project/imsearch/ADL97/.