Chapter 5

FEATURES IN CONTENT-BASED IMAGE RETRIEVAL SYSTEMS: A SURVEY

Remco C. Veltkamp, Mirela Tanase, Danielle Sent
Utrecht University, Department of Computing Science, Utrecht, The Netherlands
{Remco.Veltkamp,mirela,danielle}@cs.uu.nl

Abstract This article provides a framework to describe and compare content-based image retrieval systems. Sixteen contemporary systems are described in detail, in terms of the following technical aspects: querying, relevance feedback, result presentation, features, and matching. For a total of 44 systems we list the features that are used. Of these systems, 35 use any kind of color features, 28 use texture, and only 25 use shape features.

Keywords: Features, CBIR systems, survey.

5.1 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 image 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. [21], [6], [7], [3], [17]. This survey however, is about the functionality of contemporary image retrieval systems in terms of the following technical aspects: querying, relevance feedback, result presentation, features, and matching. This article is a short version of a much more comprehensive survey [24], which covers substantially more systems, and also treats the following aspects: indexing data structures, performance, and applications.

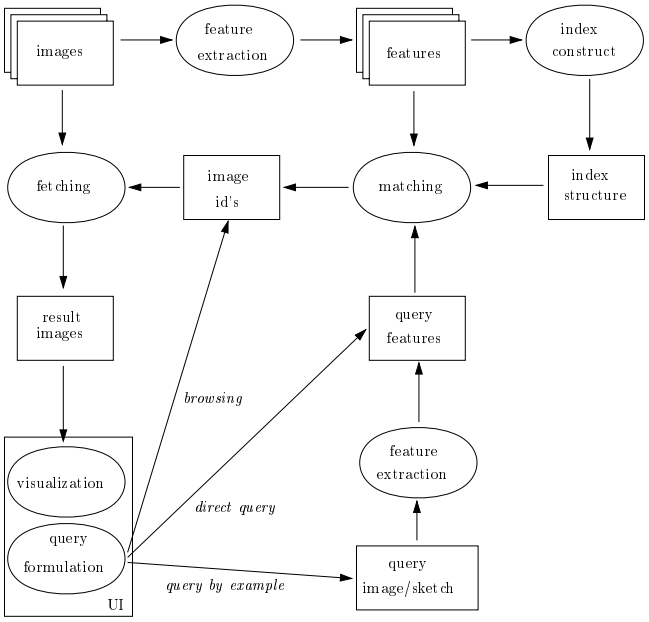

Figure 5.1 Content-based image retrieval framework.

A number of keyword-based general WWW search engines allow to indicate that the media type must be images, such as HotBot (`http://hotbot.lycos.com/`), and NBCi (`http://www.nbci.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 picture searcher of Lycos (`http://multimedia.lycos.com/`), but 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 (`http://image.altavista.com/cgi-bin/avncgi`) 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 other such content-based image retrieval systems.

5.2 Criteria

Many image retrieval systems can be conceptually described by the framework depicted in figure 5.1. 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, 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 way 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 interaction models is given in [25]. 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, texture, 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 has been allusive, 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. 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 [23]. 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.

5.3 Systems

Below we describe a number of content-based image retrieval systems, in alphabetical order.

1 Blobworld Dept. Computer Science, University of California, Berkeley. `http://elib.cs.berkeley.edu/photos/blobworld/`. [2].

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, as the 2D coordinate $(contrast, contrast \times anisotropy)$. Shape is represented by (approximate) area, eccentricity, and orientation.

Figure 5.2 Blobworld.

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 coordinates in representation space. The distance between centroids is the Euclidean distance. The distances are combined into a single final distance.

Result presentation The retrieved images are ranked in linear order, and presented together with the segmented version showing the regions, see figure 5.2.

2 C-bird (Content-Based Image Retrieval from Digital libraries). School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada. `http://jupiter.cs.sfu.ca/cbird/`. [11].

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

Figure 5.3 C-bird.

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 illumination invariance, one of the buttons under the selected query image is pressed (see figure 5.3, left). For a query by color or texture layout, grids are presented for drawing color, texture density and edge orientation layout (see figure 5.3, 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 invariant 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.

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).

3 ImageMiner. Technologie-Zentrum Informatik, Univerity of Bremen, Germany. `http://www.tzi.de/bv/ImageMinerhtml/`. [9].

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 classify scenes and generate a description with keywords and values.

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

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

4 ImageRETRO (Image RETrieval by Reduction and Overview). Department of Computer Science, University of Amsterdam, The Netherlands. `http://carol.wins.uva.nl/~vendrig/imageretro/`. [25].

Querying The user is presented with 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 color 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.

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 devided 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.

5 ImageRover. Dept. Computer Science, Boston University. `http://www.cs.bu.edu/groups/ivc/ImageRover/`. [22].

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: central, upper right, upper left, lower right, lower left). 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 he is looking for. Apart from selecting relevant images, the user can deselect 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.

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.

6 LCPD (Leiden 19th Century Portrait Database). Dept. of Computer Science, Leiden University, The Netherlands. `http://ind156b.wi.leidenuniv.nl:2000/`. [10].

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

Figure 5.4 LCPD.

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 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 average pixel to pixel difference in the intensity or gradient space.

Result presentation Retrieved images are presented in decreasing similarity order, see figure 5.4).

7 MARS (Multimedia Analysis and Retrieval System). 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. `http://www-db.ics.uci.edu/pages/research/mars.shtml`. [16].

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 devided 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

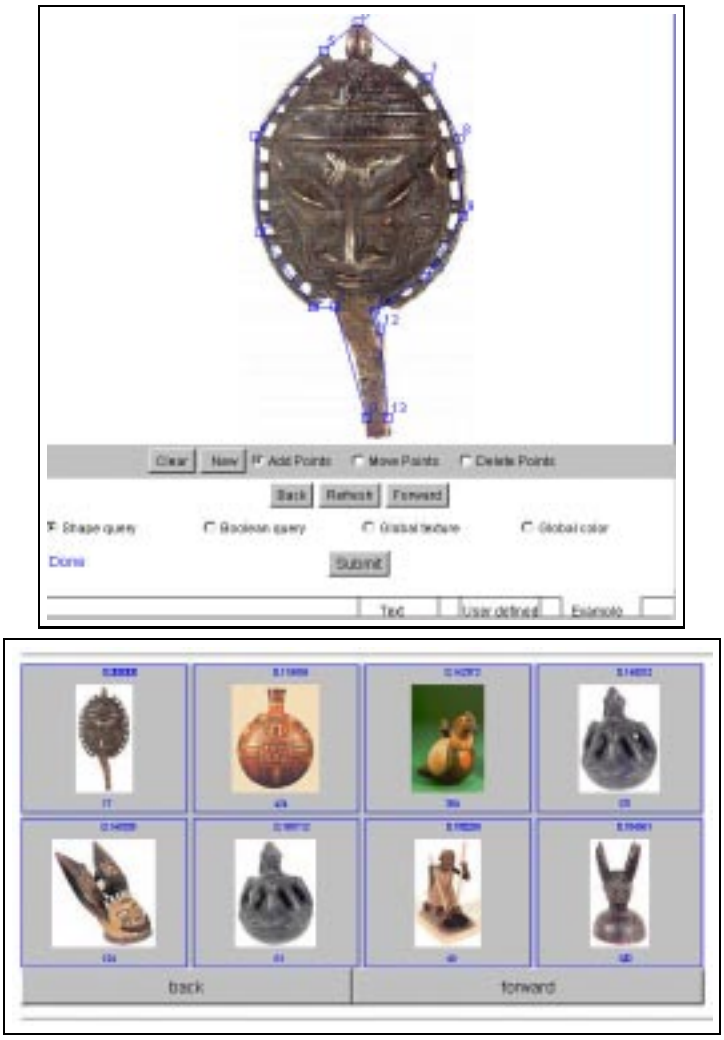

Figure 5.5 Mars.

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 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.

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.

8 MIR (Multimodal Information Retrieval System). Center of Excellence for Document Analysis and Recognition, University at Buffalo, NY, USA. `http://www.cedar.buffalo.edu/MMIR/`. [19].

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 correllation 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.

9 NETRA. Electrical and Computer Engineering, University of California, Santa Barbara. `http://maya.ece.ucsb.edu/Netra/`. [12].

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

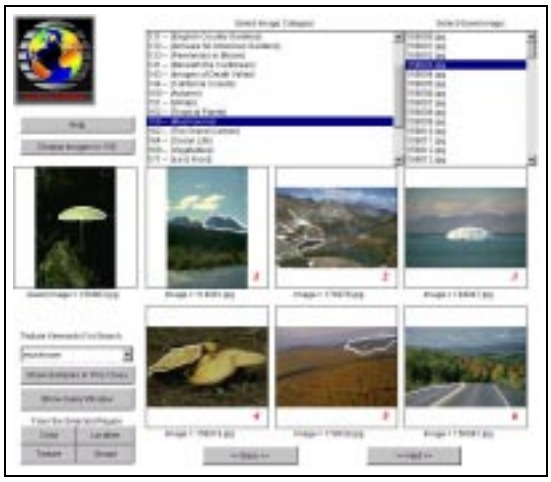

Figure 5.6 NETRA. Result of querying on shape with the complex coordinate function.

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: $f_K = (|F_1|, \ldots, |F_{32}|)$, $f_R = (|F_1|, \ldots, |F_{32}|)/|F_0|$, $f_Z = (|F_{-31}|, \ldots, |F_{-1}|, |F_2|, \ldots, |F_{32}|)/|F_1|$.

Querying There are 2,500 images 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 of 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 be-

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

Result presentation The matched images are linearly ordered, see figure 5.6.

10 Picasso. Visual Information Processing Lab, University of Florence, Italy. `http://viplab.dsi.unifi.it/PICASSO`. [1].

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. 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. 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).

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. 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, 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 2D 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 $E - M$.

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.

11 PicToSeek. Dept. Computer Science, University of Amsterdam, The Netherlands. `http://www.science.uva.nl/research/isis/pictoseek/`. [5].

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. 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. 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 histogram intersection $d(H_j^Q, H_j^I) = (\sum_{k=1}^{n} min\{H_j^Q(k), H_j^I(k)\})/(\sum_{k=1}^{n} H_j^Q(k))$, where $j \in \{A, B, C, D, E, F\}$ and n is the number of nonzero bins in H_j^Q.

Result presentation The retrieved images are shown without explicit order.

12 QBIC (Query By Image Content). IBM Almaden Research Center, San Jose, CA. `http://wwwqbic.almaden.ibm.com/`. [15].

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 $C = [c_1 c_2 \ldots c_n]$ 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 [20].

The shape features consist of shape area, circularity, eccentricity, major axis orientation and a set of algebraic moment invariants. For the database images, these shape features are extracted for all the object contours, semi-automatically 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.

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.

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.

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

13 Surfimage. INRIA Recquencourt, France. `http://www-rocq.inria.fr/cgi-bin/imedia/surfimage.cgi`. [14].

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, 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, where k_1, k_2 are the prinicpal curvature values.

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 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 non-relevant examples from the user in a statistical approach that estimates the distribution of relevant images.

14 SYNAPSE (SYNtactic APpearance Search Engine). Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA. `http://hobart.cs.umass.edu/~mmedia/`. [13].

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 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

Figure 5.7 Synapse.

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 5.7 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 $\max_m S_m$.

Result presentation Images are shown 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.

15 VisualSEEk. Center for Telecom Research, Columbia University, NY. `http://www.ctr.columbia.edu/VisualSEEk/`. [18].

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

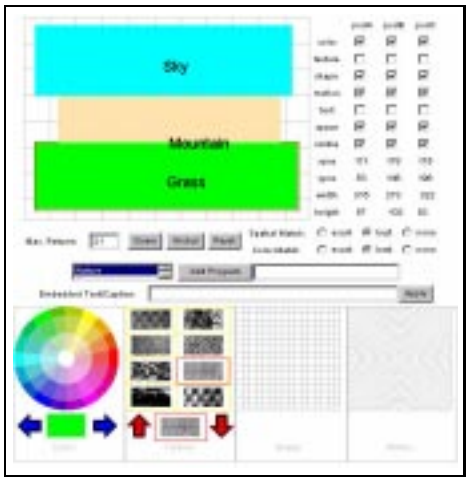

Figure 5.8 VisualSEEk.

similar regions.
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 5.8) 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.

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

16 WISE (Wavelet Image Search Engine). Department of Computer Science, Stanford University. `http://www-db.stanford.edu/~wangz/project/imsearch/`. [26].

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 (FWT) using Daubechies wavelets is applied. From the resulting 128×128 matrices, denoted 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 cri-

terion, $(\sigma^q_{C_1}\beta < \sigma^i_{C_1} < \sigma^q_{C_1}/\beta) \vee ((\sigma^q_{C_2}\beta < \sigma^i_{C_2} < \sigma^q_{C_2}/\beta) \wedge (\sigma^q_{C_3}\beta < \sigma^i_{C_3} < \sigma^q_{C_3}/\beta))$, 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 step are compared to the query using a weighted Euclidean distance between wavelet coefficients: $\sum_{m=1}^{2}\sum_{n=1}^{2} w_{mn} \sum_{i=1}^{3}(w_{C_i}\ d_E(W^i_{C_i,m,n}, W^q_{C_i,m,n}))$, with d_E the Euclidean distance, and where $W_{C_i,1,1}$, $W_{C_i,1,2}$, $W_{C_i,2,1}$, $W_{C_i,2,2}$ denote the 8×8 upper left corner, and, respectively, the three surrounding 8×8 submatrices of W_{C_i}. 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.

5.4 Summary

The features used in the 16 systems discussed in this article, and 28 more systems, are listed in table 5.1. 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 [20], 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. Using color histograms per fixed subimage is a simple way of exploiting spatial layout; considering the relative position (constellation) of regions is more complex. See [24] for what the 'other' features are.

Of the 44 systems in the table, 35 use any kind of color features, 28 use texture, 25 use shape, 16 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, so that only overly simple features such as area are used, which is often little effective.

		ADL	AltaVista	Amore	ASSERT	BDLP	Blobworld	CANDID	C-bird	CBVQ	Chabot	CHROMA	DrawSearch	Excalibur	FIDS	FIR	FOCUS	ImageFinder	ImageMiner	ImageRETRO	ImageRover	ImageScape	Jacob
Keywords		⋆		⋆		⋆					⋆										⋆		
Face Detection																						⋆	
Layout					⋆		⋆		⋆			⋆			⋆		⋆		⋆		⋆	⋆	
Shape	No details available		⋆					⋆						⋆									
	Other								⋆											⋆			
	Elementary descriptors				⋆		⋆		⋆	⋆							⋆		⋆			⋆	
	Bounding box/ellipse																		⋆				
	Curvature scale space																						
	Elastic models																						
	Fourier descriptors												⋆									⋆	
	Template matching			⋆																			
Texture	No details available		⋆					⋆						⋆				⋆					
	Other																						
	Edge statistics				⋆				⋆						⋆						⋆		⋆
	Local binary patterns														⋆							⋆	
	Random fields												⋆										
	Atomic texture features				⋆		⋆												⋆		⋆	⋆	⋆
	Wavelet, Fourier transform	⋆								⋆					⋆	⋆							
Color	No details available		⋆					⋆						⋆				⋆					
	Other				⋆	⋆			⋆											⋆		⋆	
	Dominant colors			⋆					⋆			⋆							⋆			⋆	
	Region histogram				⋆		⋆																
	Fixed subimage histogram														⋆		⋆				⋆		
	Color coherence vector																						
	Average colors				⋆								⋆							⋆			
	Correllation histogram																						
	Global histogram								⋆	⋆	⋆				⋆					⋆	⋆		⋆
	Eigen image																						
System																							

Table 5.1 Summary of systems and features.

		LCPD	MARS	MetaSEEk	MIR	NETRA	Photobook	Picasso	PicHunter	PicToSeek	QBIC	Quicklook	SIMBA	SQUID	Surfimage	SYNAPSE	TODAI	VIR	VisualSEEk	VP	WebSEEk	WebSeer	WISE
Keywords			⋆	⋆	⋆				⋆		⋆						⋆	⋆			⋆	⋆	
Face Detection					⋆		⋆															⋆	
Layout			⋆		⋆	⋆		⋆			⋆	⋆							⋆				
Shape	No details available									⋆								⋆					
	Other																			⋆			
	Elementary descriptors							⋆			⋆	⋆							⋆				
	Bounding box/ellipse							⋆											⋆				
	Curvature scale space													⋆									
	Elastic models						⋆	⋆							⋆								
	Fourier descriptors		⋆			⋆																	
	Template matching										⋆												
Texture	No details available																	⋆					
	Other	⋆											⋆		⋆	⋆							
	Edge statistics											⋆			⋆		⋆						
	Local binary patterns	⋆																					
	Random fields						⋆																
	Atomic texture features		⋆	⋆							⋆	⋆			⋆								
	Wavelet, Fourier transform		⋆			⋆						⋆			⋆								⋆
Color	No details available																	⋆					
	Other											⋆										⋆	
	Dominant colors																		⋆				
	Region histogram					⋆		⋆															
	Fixed subimage histogram		⋆																				
	Color coherence vector								⋆			⋆											
	Average colors										⋆												
	Correllation histogram				⋆				⋆			⋆											
	Global histogram		⋆	⋆	⋆				⋆	⋆	⋆	⋆			⋆						⋆	⋆	
	Eigen image						⋆								⋆								

Table 5.1 (Continued) Summary of systems and features.

5.5 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 technique, 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.

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 classifies landscapes, and body plans have been used to recognize animals and people [4]. Once separate entities in images are recognized, the way to semantic reasoning lies open.

References

[1] 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.

[2] 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 [8].

[3] 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`.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

[10] Michael S. Lew, D. P. Huijsmans, and Dee Denteneer. Content based image retrieval: KLT, projections, or templates. pages 27–34. Amsterdam University Press, August 1996.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Rohini Srihari, Zhongfei Zhang, and Aibing Rao. Intelligent indexing and semantic retrieval of multimodal documents. *Information Retrieval*, 2(2):245–275, 2000.

[20] 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.

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

[22] 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.

[23] Remco C. Veltkamp and Michiel Hagedoorn. State-of-the-art in shape matching. In Michael Lew, editor, *Principles of Visual Information Retrieval*. Springer, 2001.

[24] Remco C. Veltkamp and Mirela Tanase. Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, Utrecht University, Department of Computer Science, October 2000. See `http://www.aa-lab.cs.uu.nl/cbirs/` for an updated version.

[25] 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, `http://carol.wins.uva.nl/~vendrig/thesis/`, February 1997.

[26] J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. 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.