지구물리 제 4권 제 3호, p. 183-195 (2001년 9월)

Content-Based Image Retrieval Techniques

Kulwinder Singh, Ming Ma, Dong-Won Park
Dept. of Information and Communications, Paichai University, Daejeon, South Korea.

ABSTRACT : This paper contains links and facts to a number of projects on "content-based access to image databases" around the world today. The main focus is on what kind of image features are used but also the user interface and the users possibility to interact with the system.

1. Introduction

With the introduction of the World Wide Web and the increased memory and processor capacity allowing storage of large amounts of digital data, the need to handle queries and browse in large image databases has arisen. Image databases exist for storing art collections, satellite images, medical images, trademark images and general collections of photographs.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. Content Based Image Retrieval (CBIR) systems have received a lot of attention from the academic and commercial development community in recent years. The aim of such systems is to enable the users pose queries such as, "retrieve images similar to a given image" from a large image database. The CBIR systems need to extract low level or high level features of an image, index them using appropriate structures and efficiently process user queries providing the required answers.

In this paper we have analyzed some technical aspects: querying, relevance feedback, features, matching, indexing data structures and result presentation of current content-based image retrieval systems.

1. Standards of CBIR

Many image retrieval systems can be conceptually described by the model depicted in figure 1.In this article we survey how the user can formulate a query, 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.

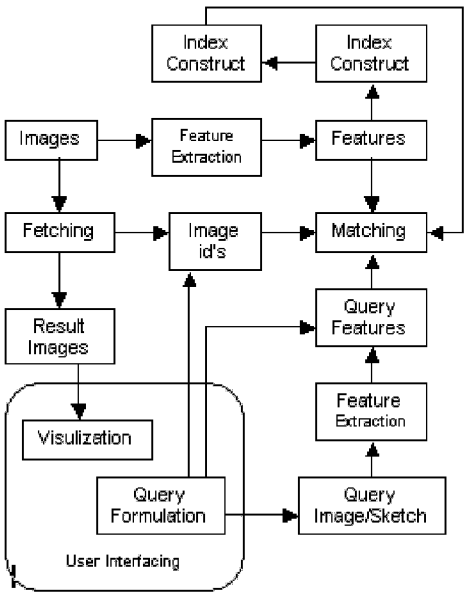

Figure 1: Content Based Image Retrieval Model

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.

In this paper, 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.

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.

2. Link to online CBIR Systems

Amore

(Advanced Multimedia Oriented Retrieval Engine)

Developer C & C Research Laboratories NEC USA, Inc.

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

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

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.

Indexing: Indexing is performed by the COIR (Content-Oriented Image Retrieval) system.

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

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.

Berkeley Digital Library Project

Developer University of California, Berkeley.

URL http://elib.cs.berkeley.edu/photos/all.shtml

Features: There are a number of alphanumerical keys available for querying: the collection, keywords, 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.

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.

Blobworld

Developer Computer Science Division, University of California,Berkeley.

http://elib.cs.berkeley.edu/photos/blobworld/start.html

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 h1 and h2, the quadratic form distance is used:

$d(h1,h2) = (h1-h2)^T A(h1-h2)$, 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 X anisotropy). The distance between centroids is the Euclidean distance. The distances are combined into a single final distance.

Indexing Rather than actually computing the distances between the full color histogram vectors of length 218 as $d(h1,h2) = (h1-h2)^T A(h1-h2)$, singular value decomposition (SVD) is used to project the histogram vectors onto a lower-dimensional subspace.

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

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

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

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 forma 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.

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/

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 regionsassociated 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. For a query by color or texture layout, grids are presented for drawing color, texture density and edge orientation layout . 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 L2 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 imagewhere 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).

Chabot

Developer Department of Computer Science, University of California, Berkeley, CA, USA.
URL **http://http.cs.berkeley.edu/~ginger/chabot.html**

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 existingtext 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 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.

CBVQ

(Content-Based Visual Query)

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

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

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 aHaar 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 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 usercan 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 : 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.

FOCUS

(Fast Object Color-based Query System)
Developer Department of Computer Science, University of Massachusetts, Amherst, MA.
URLhttp://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

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

Features: Each image is divided in cells of 100x100 pixels and for each cell a color histogram in the HSV space, coarsely quantized along the saturation and value axes (64x10x10), 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 increasingorder 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 L1 distances between each query peak and the matched candidate peak. To match theSPG 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.

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, theretrieved 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.

ImageScape

Developer Department of Computer Science, Leiden University,TheNetherlands.

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

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 .

Features: Edge maps of the images collected byWeb crawlers are obtained using the Sobel operator and a Gaussian blurring filter. A frequencyhistogram of the 3x3 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. 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 L1 distance on the trigram vectors to retrieve the top1% matches from the entire database. Among these, 20 matches are selected in a second step, a 20x20 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.

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

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 mx 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 3x 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 frequencyrange). 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 valueand counting the number of occurrences of each of the 27 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 L1 distance. If two images have different sizes, the similarity score between two projection vectors is given by the minimum L1 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.

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.

MetaSEEk

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

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

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 performancedatabase 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 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. This information is used to modify the corresponding entries in the performance database.

4. Conclusions

Most systems are products of research, and therefore emphasize one aspect of content-based retrieval. Sometimesthis 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 showsmore 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 surprising results.

Apparently the shape features used for matching are not the most efiective 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 efiectiveness, efficiency, and exibility. 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 aw 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 morechance 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.

Reference:

[1] N. S. Chang and K. S. Fu, A Relational Database System for Images, Technical Report TR-EE 79-28, Purdue University, May 1979.

[2] N. S. Chang and K. S. Fu, Query-by pictorial-example, IEEE Trans. on Software Engineering SE-6(6), 1980.

[3] S.-K. Chang, Pictorial data-base systems, IEEE Computer, 1981.

[4] S.-K. Chang, C. W. Yan, D. C. Dimitroff, and T. Arndt, An intelligent image database system, IEEE Trans.Software Eng. 14(5), 1988.

[5] H. Tamura and N. Yokoya, Image database systems: A survey, Pattern Recognition 17(1), 1984.

[6] S.-K. Chang and A. Hsu, Image information systems: Where do we go from here? IEEE Trans. on Knowledge and Data Engineering 4(5), 1992.

[7] Kato T "Database architecture for content-based image retrieval" in Image Storage and Retrieval Systems (Jambardino, A A and Niblack, W R, eds), Proc SPIE 1662, 112-123,1992.

[8] Santini, S and Jain, R C"The graphical specification of similarity queries" Journal of Visual Languages and Computing 7, 403-421,1997.

[9] C. S. McCamy, H. Marcus, and J. G. Davidson, A color-rendition chart, Journal of Applied Photographic Engineering 2(3), 1976.

[10] M. Miyahara, Mathematical transform of (r,g,b) color data to munsell (h,s,v) color data, SPIE Visual Commun. Image Process. 1001, 1988.

[11] J. Wang, W.-J. Yang, and R. Acharya, Color clustering techniques for color-content-based image retrieval from image databases, in Proc. IEEE Conf. on Multimedia Computing and Systems, 1997.

[12] Swain, M J and Ballard, D H"Color indexing" International Journal of Computer Vision 7(1), 11-32,1991.

[13] M. Ioka, A Method of Defining the Similarity of Images on the Basis of Color Information, Technical Report RT-0030, IBM Research, Tokyo Research Laboratory, Nov. 1989.

[14] W. Niblack, R. Barber, and et al., The QBIC project: Querying images by content using color, texture and shape, in Proc. SPIE Storage and Retrieval for Image and Video Databases, Feb. 1994.

[15] M. Stricker and M. Orengo, Similarity of color images, in Proc. SPIE Storage and Retrieval for Image and Video Databases, 1995.