# Content Based Image Retrieval Systems

John M. Zachary, Jr. and Sitharama S. Iyengar
Louisiana State University
Department of Computer Science
Baton Rouge, Louisiana

**Abstract**

*The ability to organize and retrieve visual information such as images and video is becoming a crucial problem for specialists and general computer users alike. Because processing visual information requires perceptual abilities not yet known to exist in computational form, the ability to retrieve visual information without human assistance is a rich, complex, and interesting problem. This paper presents the problem from the point of view of real-world system construction, discusses the main feature extraction methods used in modern CBIR systems, and outlines several CBIR system implementations.*

## 1. Introduction

### 1.1 Motivation

As information technology proliferates throughout our society, digital images and video, or *visual objects* are becoming as important as traditional textual based information. This phenomenon has several reasons: demilitarization of imaging and satellite technology, the emergence of the World Wide Web as a digital communications infrastructure, the impending convergence of computers and television, and the increase in use and availability of digital cameras and video recorders. Coupled with the object-oriented paradigm, these factors are advancing the idea that digital imagery and video are first-class data types. Technologies such as the Java 2D API and Apple's QuickTime API are steps in this direction. With the massive growth in the amount of visual information available, there exists a real need for systems to catalog and provide retrieval from digital image and video libraries.

Preceding a discussion of information retrieval, the nature of the information must be understood. A visual object contains two types of information: metadata and visual features. *Metadata*, or textual attributes about the object, are extracted with human assistance and stored as an accessory to the visual object. Visual features represent implicit information in the object and are derived through feature extraction algorithms.

Information retrieval is the process of converting a request for information into a meaningful set of references. In the past, "information retrieval" has meant "textual information retrieval", but the definition is equally valid when applied to "visual information retrieval". However, there is a distinction between the type of information and the nature of the retrieval of text and visual objects. Textual information can be regarded as a one dimensional array of tokens and words, but images are a two dimensional array of information and videos are three dimensional (one dimension is time) [9]. Moreover, textual retrieval is based on discovering semantic and/or syntactic similarity. Image and video is instead based on perceptual similarity, thus making VIR harder because capturing the perceptual meaning of an image by automated methods is still an open problem in machine vision and image understanding research.

Visual information retrieval plays a significant role in many application areas, such as remote sensing and satellite image databases, geographic information systems, diagnostic medical image databases, military surveillance and tracking, journalism data management, and general consumer use.

### 1.2 Outline of article

In the next section, we will discuss the fundamental aspects of VIR research. Section 3 discusses specific techniques for building feature representations. In Section

4, a few specific systems that have been built will be discussed. We conclude the paper in Section 5.

## 2. Fundamental aspects of CBIR

VIR research is concerned with answers to three basic questions [5]:

*In the general context, what is the information content of a visual object?*

Previous research has taken two approaches to solutions for the VIR problem based on the form of the visual information. Attribute-based methods rely on traditional textual information retrieval and DBMS methods as well as human intervention to extract metadata about a visual object and couple it together with the visual object as a textual annotation. Feature-based methods, on the other hand, apply image processing algorithms to a visual object to extract features that are thought to characterize the content of the visual object and form an alternative representation of the visual object amiable to query processing.

*How can a user succinctly specify a query for a piece of visual information?*

There are many ways one can pose a visual query. A good query method will be natural to the user as well as capturing enough information from the user to extract meaningful results. The following query methods are commonly used in VIR research:

a. Query-By-Example (QBE)

In this type of query, the user of the system specifies a target query image upon which the image database is to be searched and compared against. The target query image can be a normal image, a low resolution scan of an image, or a user drawn sketch using graphical interface painting tools. A prime advantage of this type of system is that is a natural way for expert and general users to search an image or video database; specialized knowledge of features, such as color distributions or regions arrangements, is not required.

b. Query-by-Feature (QBF)

In the QBF type system, users specify queries by explicitly specifying the features they are interested in searching for. For example, a user may query an image database by issuing a command to "retrieve all images whose left quadrant contains 25% yellow pixels". This query is specified by the use of specialized graphical interface tools. Specialized users of an image retrieval system may find this query type natural, but general users may not. QBIC ([3],[13]) is an example of an existing CBIR system that uses this type of query method.

c. Attribute-based queries

Attribute-based queries use the textual annotations pre-extracted by human effort as a primary retrieval key. This type of representation entails a high degree of abstraction which is hard to achieve by fully automated methods because visual objects contain a large amount of information which is difficult to summarize using a few keywords. While this method is generally faster and easier to implement, there is an inherently high degree of subjectivity and ambiguity present.

Which query method is most natural? To the general user, probably Attribute-Based queries are with QBE systems a close second. Users should expect to query VIR systems by asking natural questions such as "Give me all pictures of Grandpa before 1980." or "Find all images on the Web with a Porsche Boxster." Mapping this natural language query to a query on image or video databases is extremely difficult to do using automated methods. The ability of computers to perform automatic object recognition on general images is still an open research problem. Most research and commercial efforts are focused on building systems that perform well with QBE methods.

*How accurate and efficient is the entire retrieval process?*

The accuracy of a VIR system is the ability of the retrieval process to deliver what the user intended. Because of the nature of the information, accuracy is both subjective and objective, and is a function of the expressiveness of the query. Systems that minimize the number of false negatives first and false positives second increase accuracy. Efficiency is a more objective measure of the entire process in terms of system responsiveness, space requirements, and usability.

Henceforth, we will restrict visual objects to being digital images, although the issues involved for the retrieval of digital video are also challenging. Additionally, we will limit our discussion of VIR to feature-based methods and adopt the acronym CBIR that stands for *Content-Based Image Retrieval.*

## 3. Feature representations

Feature extraction and representation is the fundamental process behind CBIR systems. As mentioned, features are properties of the image extracted with image processing algorithms, such as color, texture, shape, and edge information. Features can be general or domain-specific; an example of a domain specific feature is weak edges extracted from satellite images to detect the presence of eddies and streams [22]. Of course, the tradeoff for deciding on a feature extraction method to use in a CBIR system is balancing between accuracy and generality. Our discussion will focus on three general features representations that have been extensively studied in the literature: color, texture, and wavelets. However, there is no single "best" feature that gives accurate results in any general setting. Usually, a combination of features is the minimally needed to provide adequate retrieval results since perceptual subjectivity permeates throughout this problem.

## 3.1 Color histograms: the foundation of CBIR

The first and most straightforward feature for indexing and retrieving images is color, the basic constituent of images (we consider grayscale a color). All other information computed by image processing algorithms starts with the color information contained in an image. The color histogram of an image is a description of the colors present in an image and in what quantities. They are computationally efficient to compute and insensitive to small perturbations in camera position. However, color histograms alone do not provide spatial information of the content in an image and are sensitive to changes in brightness, contrast, and compression artifacts. Despite their drawbacks, color histograms are useful for quick indexing into large image databases and as a foundation for more advanced feature representations.

Color in images is typically represented in the RGB color model. The size of each channel is 8 bits or 256 levels of each channel. Thus, there are $2^{24}$ or 16,777,216 possible colors available. Clearly, it is counterproductive to computer the histogram for each possible color, especially when one considers that the difference between two RGB triplets that differ by a small magnitude in one channel may be barely perceptible to the human observer. In fact, from a perceptual point of view, the RGB model is not the best; it is a model appropriate to hardware devices such as raster displays. However, because of it prevalence and simplicity, we limit our discussion to it as the basis for color histograms. For a discussion of alternative color models, the reader is referred to Gonzales and Woods [4].

In order to compute the color histogram of an image, the color model of the images must first be discretized to contain n colors. A method for doing this is to consider the k most significant bits from each channel ($k \leq 8$). Thus, there are $2^{3k}$ distinct colors in which to classify the pixels of an image. In our research, we use k values of 2 and 3, giving us 64 or 256 bin histograms.

The vector

$$H(M) = \langle h_1, h_2, \ldots, h_n \rangle$$

is the n-dimensional feature vector representation of the color histogram for image M where $h_j$ is the number of pixels of color j in M. An alternative definition that we use is $h_j$ is the proportion of pixels of color j in M relative to the total number of pixels in M. This definition allows us to use arbitrary sized images.

This definition captures the distribution of color in image only; shape, texture, and other image properties are totally lost. One can easily imagine an example of two images that are dissimilar to any human observer yet will have identical color histograms. For example, the picture of a red ball on a black background will have the same histogram that an image with the same number of red and black pixels randomly distributed has.

To compare two histograms, $H_1 = H(M_1)$ and $H_2 = H(M_2)$, the distance between them is computed. Analytic metrics, such as the $L_1$-norm, $L_2$-norm or $L_\infty$-norm, are typically used. While these metrics follow from the vector space definition of color histograms, they are inflexible to artifacts as color shifting, changes in registration, and phenomena related to image formats, such as dithering and compression artifacts [10].

## 3.2 CCVs and CCV-TEVs: augmenting color histograms

Stricker and Swain [26] present a detailed analysis of color histograms as indexing mechanisms. They make the observation that indexing by color histograms works if the histograms are sparse, i.e. most of the images contain only a fraction of the colors. The implication is images that contain all of the same colors or images that contain a large percentage of the color available in the color model will likely have histograms that are close together in the feature vector space. Because of the main drawback of color histograms, many research efforts have focused on hybrid representations in which additional properties of images augment or replace the color histogram. It can be said that for any image property that can be computed by an image processing algorithm, a new feature vector can be "discovered". This paper will present only a small sample of the many augmented color histogram methods in the literature. The bottom line is that any of these augmented feature vectors have their place depending on the nature of the images in the database.

A color histogram enhancement proposed by Pass, Zabih, and Miller [15] augments color histograms with spatial information. *Color Coherence Vectors* (CCVs) classify each pixel as either coherent or incoherent based on whether it is part of a large color-homogeneous region in the image. After the classification, the histogram is constructed where each bucket is a color (as in the original color histogram formulation) and the value associated with each bucket is the number of coherent pixels. The basic CCV algorithm is given by

**begin (CCV)**

Step 1. Blur the image with a low pass 3x3 filter to remove local noise and discretize the color model into n distinct colors.

Step 2. Classify pixels into buckets.

> 2(a) Compute the connected components in the image. A connected component C is a maximal set of pixels in the image such that for any two pixels p and q, there is a connected path from p to q in C.
>
> 2(b) A given pixel is coherent if the number of pixels of the connected component to which it belongs is greater than some user defined constant $\tau$.

**end (CCV)**

The representation of a CCV is

$$H_{CCV}(M) = <(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)>$$

where $\alpha_j$ is the number of pixels in a connected component of color j and $\beta_j = h_j - \alpha_j$. Comparisons of these features vectors is based on the same analytic metrics used in basic color histograms.

One shortcoming of the CCV method is that it does not capture the relationship of a connected component to its background. It also fails to capture the shape of the connected component. A connected set of pixels that twists around in an image is not discriminated against one that has a sense of 'blobiness' to it. We have developed an augmentation to the CCV method that simultaneously addresses these two issues. Our method augments CCVs by storing an additional vector containing edge information we term Threshold Edge Vector (TEV). Hence, our augmented method is termed CCV-TEV.

A simple method for measuring edge information is to enumerate the edge pixels in each of the connected components of the CCV algorithm. However, the CCV method relies on color model discretization that destroys the weak edges present in an image and disrupts the edges in connected components of size less than $\tau$. Thus, we choose to compute gradient information directly from the original image using the well-known Sobel operators [4]. Our algorithm is given by

**begin (CCV-TEV)**

Step 1. Given an image I with colors represented in the RGB color model, perform a low pass filter over the image to remove noise and compute the CCV using the previous algorithm.

Step 2. Convert I to a grayscale image $I_y$ of intensity values using the formula

$$y = 0.299*r + 0.587*g + 0.114*b.$$

Step 3. Use the Sobel operators to transform $I_y$ to the gradient image $I_\nabla$ whose values will be in the range [0, 255].

Step 4. Given a constant v in the range (0,255), compute the number of pixels in $I_\nabla$ with intensity value j where j=v, v+1,..., 255. Denote this vector

$$H_{TEV} = < g_v, g_{v+1}, \ldots, g_{255}>.$$

**end (CCV-TEV)**

This algorithm will produce a vector of length 255-v for each image and classifies it based on the strength of the gradients in the image. The new CCV-TEV representation is

$$H_{CCV+}(M) = <H_{CCV}, H_{TEV}>$$

which is analytic similar to the CCV and basic color histogram. Thus, the typical norms for comparison apply.

The lower bound v is defined because in many images a large portion of $I_\nabla$ will be values near or equal to 0. By establishing a minimum threshold value v, the TEV histogram is not dominated by useless intensity values, but weak edges can be captured. In some applications, the presence of weak gradients indicates important physical phenomena [22]. TEVs succinctly describe the distribution of weak edges in an image and prevent pixels of strong gradients in one image from matching the pixels of weak gradients in another image. The $H_{CCV+}$ representation is a constraint-based version of the $H_{CCV}$ in the sense that similar images should have similar gradient patterns as well as similar regions of connectedness.

Preliminary results indicate that our new image representation acts as an additional filter to the CCV method. While not perfect, our method returns slightly better results than the CCV method, especially for images containing high amounts of gradient information such as scenes of natural imagery.

### 3.3 Texture

Texture is ubiquitous; it can be hard to define, but we know it when we see it. Ironically, it is a well-studied phenomenon in the areas of machine vision, pattern recognition, and image classification ([4],[8],[17]). Picard [18] discusses three properties of texture

*Texture lacks a specific complexity.*

Picard illustrates this property by considering three categories of patterns, illustrated by the strings in Table 1. The periodic texture string is characterized by having a basic primitive, replication rules, and a tolerance for errors in the string. The basic complexity of the substring "HelloWorld" is lost as the size of the string grows. The stochastic texture string lacks any basic rules for construction except for the randomness upon which each subsequent token is chosen. White noise generated from a Gaussian distribution is a basic example of this type of texture. The final string is a permutation of the second into a form having some semantic meaning. The reason it is not considered a texture is due to its specificity. Although it chosen from the same probability distribution as the second, it's specific order makes it different. Cars, faces, and other objects in images that humans excel at recognizing while pattern recognition researchers continue to toil over are examples of this type of non-texture element.

| | |
|---|---|
| Periodic Texture | HelloWorldHellloWorldHelloWorld |
| Stochastic Texture | gshy ldsMaoef a |
| Non-Texture | My dog has fleas |

**Table 1. String patterns illustrating texture.**

*Texture contains high frequency information.*

High frequency image information tends to occur very often in texture although they may also occur in non-textures. Algorithms to detect edges, motion, and compute wavelet decompositions exploit the presence of this type of information in an image.

*Texture has a finite range of scalability.*

Textures do not tend to be defined on a large range of scales and have fractal dimensions associated with them. When observed from afar, highly textured surfaces tend to appear smooth or periodic. It isn't until the distance between the observer and the textured object is small relative to the texture that the observer begins to detect it. For example, grass appears green and smooth when viewed from a high altitude. It isn't until one is close relative to the size of the blades of grass that one can detect the textured nature of grass.

There are three primary approaches to describing texture [4]. A statistical approach developed by Haralick et al. [8] is the gray level co-occurrence matrix. This method characterizes texture by generating statistics of the distribution of intensity values as well as the position and orientation of similar valued pixels. A structural approach to texture representation is characterized by generating complex texture patterns from lower level texture primitives, similar to how regular languages are generated by finite state automata. The third, and perhaps the most recently popular, method for texture description is through spectral methods. The use of Fourier and Wavelet spectral methods for texture description and classification has received widespread attention in the literature ([1],[4],[11],[20]).

### 3.4 Wavelets

In addition to texture description, wavelets have also been used in a wider capacity for CBIR. Jacobs et al. [10] used wavelets to search an image database from a low resolution version or user drawn sketch of the target image. Their approach is to create image signatures of each image in the database as well as the example image from the Haar wavelet decomposition of the images. Each signature is a truncated and quantized version of the coefficients. A query is performed by determining how many significant coefficients the example image has in common with the signatures of the images in the database.

We are currently investigating the use of a different wavelet transform to emphasize the presence of weak edges as an important feature representation [22]. In some application domains, such as oceanography and remote sensing, the presence of weak edges indicates an important physical phenomenon. The Starck-Murtagh-Bijaoui wavelet transform has successfully been applied to extract this feature from oceanographic imagery. Our approach is to construct multidimensional indices from the computed edge information.

## 4 CBIR systems

CBIR systems generally display a dichotomy between the degree of automated feature extraction and the level of dependence on domain knowledge. Gudivada and Raghavan [7] describe systems that can compute needed visual features with human assistance as *dynamic feature extraction* systems. The approach that achieves a reasonable level of generality at the expense of automated feature extractions is called *a priori feature extraction.* In the following discussion, we briefly describe some of the well-known CBIR systems that have been developed.

## 4.1 IBM's QBIC

One of the best known CBIR systems is QBIC (Query by Image Content) ([3],[13]), developed at IBM and integrated in several products. QBIC allows queries to be made on large image and video databases. Queries in QBIC are based on a hybrid approach, combining color, texture, shape, user sketches, and spatio-temporal features, and are specified by a rich set of graphical user interface tools. The structure of QBIC is organized into two main components: the database population and the database query. The database population stage processes images and video to extract visual features and form an index for database storage and retrieval, which makes it an *a priori* system. After the user has graphically composed the query, the database query component extracts features from the query and computes a distance metric on the indices in the database. One of the prime advantages of QBIC is its rich set of graphical user interface tools for composing queries. A demonstration of the QBIC system is available on the WWW at http://wwwqbic.almaden.ibm.com.

## 4.2 WebSeek

WebSeek [24] is another well-established system with the goal of providing a directory of visual objects available on the World Wide Web. The system uses a hybrid attribute/feature based approach to indexing and query construction of the visual objects it accumulates. WebSeek uses autonomous agents ("spiders") to accumulate, process, and catalog visual objects.

- The *Traversal Spider* finds candidate WWW pages with embedded or links to visual objects by performing a breadth-first search from a root URL. The HTML of the candidate page is then passed to the *Hyperlink Parser*.
- The *Hyperlink Parser* parses HTML in search of URLs that suggest possible visual object content. Multipurpose Internet Mail Extension (MIME) labels are used to map the URL file extension to a possible visual object type. The list of visual object URLs are then passed to the *Content Spider*.
- The *Content Spider* retrieves the visual object from the URL location, computes an index from extracted features and associated metadata (width, height, visual object type, etc.), and places the index and generated thumbnails in the image database.

The end result is a catalog of visual objects upon which users may search through the WWW. There are several important issues involved with WebSeek's classification, search, and retrieval processes beyond the scope of this paper. We are, however, interested in the content-based methods of the system. WebSeek incorporates two content-based methods: a global feature query based on color histograms and a query based on the spatial arrangement of color regions. The color histogram is composed of 166 bins in HSV space. The indexing method is a binary tree method that increases the retrieval process. Color histograms do not capture locally important properties of images such as the spatial location and arrangement of color regions. Thus, WebSeek provides the capability to query this type of local image property. This integrated spatial and color feature query feature that allows the user to place color regions on a grid is a refinement of the color histogram space [23].

## 4.3 MIT's Photobook

The MIT Media Laboratory has developed a system called Photobook [16] that provides interactive tools designed for browsing and retrieval from image and video databases. At the heart of Photobook is a "select-sort-redisplay" process in which the first step is based on the user selecting an image category from which the system retrieves a subset of the image database by matching textual annotations against the category. The Photobook graphical interface presents these images to the user using a photograph album paradigm. From this, the user chooses an image or set of images upon which Photobook sorts the image set based on similarity to the selected image(s). The set is re-presented to the user and the cycle is repeated to refine the user's search.

Image similarity is computed using three classes of image features. Photobook can compute similarity based on shape, texture, or appearance applied to facial recognition. The three can also be combined with one another as well as incorporating the textual annotations available in the first step of the select-sort-redisplay process.

An extension to Photobook called FourEyes has been developed recently to provide interactive user tools that can learn appropriate features based on user supplied examples [17], [19]. The rationale of the extension is that no single of small collection of feature extraction methods is sufficient to compute similarity for an arbitrary set of images. The computational model for FourEyes is based on statistical learning theory and the concept of a "society of models" [18].

### 4.4 NETRA

NETRA [12] is a prototype image retrieval system developed at the University of California at Santa Barbara. The system uses a hybrid approach to feature extraction by incorporating color, texture, and shape information from an image in its indexing method. The characteristic feature of NETRA is it uses segmented local regions for indexing images in the database. Thus, both global and local characteristics are exploited.

### 4.5 Chabot

The Chabot system [14] is another a priori system with the goal of integrating feature extraction behavior within a traditional RDBMS to handle large scale image databases. The feature extraction analysis is based on color histogram representations of images and query optimization algorithms. The main goals of Chabot are (a) integration of data types, (b) scalability and a multiple level storage plan, (c) simplicity of use, (d) flexible query methods, and (e) querying by image content. The Chabot system has been tested and refined on an image database of over 500,000 images.

## 5 Future Research Directions

Content-based image retrieval research is indeed a large tree bearing much low-hanging fruit. Several problems have not been sufficiently addressed in the short period this area has received research attention.

- Large-scale image databases are result of the growth of the WWW as well as the increase in capacity/decrease in cost of secondary storage devices. However, as feature extraction methods grow in sophistication and dimension it is clearly infeasible to search an index space sequentially while searching for similar images to retrieve. Dimension reduction of the feature space and sophisticated multidimensional indexing schemes are two approaches to this problem.

- New knowledge representation advances to bridge the gap between low-level image features and high-level perceptual objectivity are needed to increase the accuracy of the retrieval process. Of course, this is the traditional focus of computer vision research. Image modeling, improved object model databases, and image processing advances are all needed. Relevance feedback research efforts are steps in this direction [2],[21].

- HCI tools must be advanced to accommodate humans better in the process. Grand visions of automated image retrieval systems will evolve to development of domain-specific interactive systems. In order to make these systems usable by non-computer vision experts, Query-By-Example must be expanded to make the query process accessible by these users.

- Study which feature extractions methods perform best for domain specific applications, e.g., medical imaging, remote sensing, and face recognition databases. There are a plethora of feature extraction methods proposed, some of which perform better than others under certain assumptions and conditions. For example, weak edge detection is an important problem in oceanographic images. Wavelet-based methods to detect these features [22] will be more important than, perhaps, in detecting object in natural scenery. FourEyes [17] is an interactive tool that allows the user to choose the feature extraction algorithm appropriate for a given task, but a complete study is needed to facilitate the development of tools for that sake of non-expert users.

## 6 Conclusion

With the explosion in multimedia capabilities over the Internet as well as on the desktop, content-based image retrieval systems are an important piece of the information retrieval landscape. There are several areas for future research not discussed. While it would be very nice indeed to have a fully automated CBIR system, it is clear that for the time being (luckily) humans are an integral part of the process. Thus, thoughtful designs for "interactive systems" should be explored. Also, a comprehensive study of features and feature combinations for various domain-specific applications such as medicine is needed. While we cannot hope to provide a discussion for every aspect of this challenging field of study, we hope the reader is sufficiently motivated to obtain some of the references and think about some of the open problems.

## 7 Acknowledgements

We gratefully acknowledge the timely response of IBM Almaden Research Center to our request for research papers concerning QBIC. We also thank Dr. Don Kraft for stimulating our interest in the area in his unique way.

## 8 References

[1] T. Chang and J. Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform", *IEEE Trans. Image Processing*, 2(4), 1993.

[2] I. Cox, M. Miller, S. Omohundro, and P. Yianilos, "PicHunter: Bayesian Relevance Feedback for Image Retrieval", *International Conference on Pattern Recognition*, Vienna, Austria, 1996.

[3] M. Flickner, H. Sawhney, W. Niblack, and et al., "Query By Image and Video Content: The QBIC Project", *IEEE Computer*, 28(9), 1995.

[4] R. Gonzalez and R. Woods, *Digital Image Processing*. Addison-Wesley, 1993.

[5] A. Gupta and R. Jain, "Visual Information Retrieval", *Comm. ACM*, 40(5), 1997.

[6] A. Gupta, "Visual Information Retrieval Technology: A Virage Perspective", Virage, Inc., 1997.

[7] V. Gudivada and V. Raghavan, "Content-Based Image Retrieval Systems", *IEEE Computer*, 28(9), 1995.

[8] R. Haralick, K. Shanmugan, and I. Dinstein, "Texture Features for Image Classification", *IEEE Trans. Systems, Man, Cybernetics*, 3(6), 1973.

[9] J. Huang, Color-Spatial Image Indexing and Applications, Cornell University Dept. of Computer Science PhD Thesis, August 1998.

[10] C. Jacobs, A. Finkelstein, and D. Salesin, "Fast Multiresolution Image Querying", *SIGGRAPH 95 Conference Proceedings*, 1995.

[11] A. Laine and J. Fan, "Texture Classification by Wavelet Packet Signature", *IEEE Trans. Pattern Recognition and Machine Intelligence*, 15(11), 1993.

[12] W. Ma and B. Manjunath, "NETRA: A Toolbox for Navigating Large Image Databases", *IEEE Intl. Conf. Image Proc.*, to appear, 1998.

[13] W. Niblack, R. Barber, W. Equitz, et al., "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape", *Proc. SPIE Storage and Retrieval for Image and Video Databases*, vol. 1908, 1993.

[14] V. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of Images", *IEEE Computer*, 28(9), 1995.

[15] G. Pass, R. Zabih, and J. Miller, "Comparing Images using Color Coherence Vectors", *ACM Conf. Multimedia*, November 1996.

[16] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases, *SPIE Storage and Retrieval of Image & Video Databases II*, San Jose, CA, pp. 34-47, 1994.

[17] R. Picard and T. Minka, "Vision Texture for Annotation", *Multimedia Systems*, 3, 1995.

[18] R. Picard, "A Society of Models for Video and Image Libraries", MIT Media Lab Perceptual Computing Section TR 360, 1996.

[19] R. Picard, T. Minka, and M. Szummer, "Modeling User Subjectivity in Image Libraries", *Proc. IEEE International Conference on Image Processing*, 1996.

[20] L. Prasad and S. Iyengar, *Wavelet Analysis with Application to Image Processing*, CRC Press, 1997.

[21] Y. Rui, T. Huang, S. Mehotra, M. Ortega, "A Relevance Feedback Architecture in Content-Based Multimedia Information Retrieval Systems", *Proc. Of IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[22] Kiran Simhadri, S. S. Iyengar, Ronald Hoyler, Matthew Lybanon, and John Zachary, "Wavelet-Based Feature Extraction from Oceanographic Images", *IEEE Transactions on Geoscience and Remote Sensing*, 36(3), May 1998.

[23] John R. Smith and Shih-Fu Chang, "VisualSeek: A Fully Automated Content-Based Image Query System", *Proceedings of the ACM International Conference on Multimedia*, 1996.

[24] John R. Smith and Shih-Fu Chang, "Visually Searching the Web for Content", *IEEE Multimedia*, 4(3), 1997.

[25] John R. Smith, "Integrated Spatial and Feature Image Systems: Retrieval, Analysis, and Compression", Columbia University, Graduate School of Arts and Sciences, February 1997.

[26] Markus Stricker and Michael Swain, "The Capacity and the Sensitivity of Color Histogram Indexing", The University of Chicago Department of Computer Science TR 94-05, 1994.