

# A Survey on: Contents Based Search in Image Databases

Björn Johansson  
Computer Vision Laboratory  
Department of Electrical Engineering  
Linköping University, SE-581 83 Linköping, Sweden  
bjorn@isy.liu.se

December 8, 2000

## **Abstract**

This survey contains links and facts to a number of projects on content based search in 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 (i.e. what 'visual language' is used).

This is a printable version of a paper in HTML format. The report may be modified at any time as will this version. The latest version of this paper is available at <http://www.isy.liu.se/cvl/Projects/VISIT-bjojo/>.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Applications</b>	<b>5</b>
2.1	General . . . . .	5
2.2	Examples . . . . .	5
<b>3</b>	<b>Some projects around the world</b>	<b>6</b>
3.1	Overview . . . . .	7
3.1.1	Image features . . . . .	7
3.1.2	User interface - visual language . . . . .	13
3.1.3	Evaluation methods . . . . .	14
3.1.4	Projects overview . . . . .	15
3.2	Details and links . . . . .	17
3.2.1	ALISA (Adaptive Learning Image and Signal Analysis) . . . . .	17
3.2.2	ASSERT (Automatic Search and Selection Engine with Retrieval Tools) . . . . .	19
3.2.3	Blobworld . . . . .	21
3.2.4	CANDID (Comparison Algorithm for Navigating Digital Image Database) . . . . .	23
3.2.5	Chabot . . . . .	24
3.2.6	Excalibur Visual RetrievalWare . . . . .	25
3.2.7	Fast Multiresolution . . . . .	26
3.2.8	ImageSearch (ImageScape) . . . . .	27
3.2.9	LCPD, Leiden . . . . .	28
3.2.10	MARS (Multimedia Analysis and Retrieval System) . . . . .	29
3.2.11	Multiresolution Image Database Search . . . . .	30
3.2.12	NETRA (means eye in Sanskrit) . . . . .	31
3.2.13	Photobook/FourEyes . . . . .	32
3.2.14	PicSOM . . . . .	33
3.2.15	QBIC (Query By Image Content) . . . . .	34
3.2.16	SIMPLiCity (Semantics-sensitive Integrated Matching for Picture Libraries) . . . . .	35
3.2.17	SQUID . . . . .	36
3.2.18	Surfimage . . . . .	37
3.2.19	Virage Image engine . . . . .	38
3.2.20	VisualSEEK/SaFe (Spatial and Feature query system) . . . . .	39
3.2.21	WebSEEK . . . . .	41
3.2.22	Huet and Hancock . . . . .	41
3.2.23	Jain and Vailaya . . . . .	42
<b>4</b>	<b>Review articles</b>	<b>43</b>
<b>5</b>	<b>Acknowledgment</b>	<b>43</b>

# 1 Introduction

With the introduction of the World Wide Web and the increased memory capacity allowing storage of large amounts of digital data, the need to handle queries and browse in large image databases has arisen. Since the beginning of the 1990's there has been an increased research activity in the area of *content based image retrieval* (CBIR). Both large research teams, for instance the QBIC project (see details below) at IBM and the Advent (All Digital Video Encoding, Networking, Transmission) project at Columbia University, New York (<http://www.ctr.columbia.edu/advent/>), and smaller project groups in the academic world and industry (see section 3) have devoted themselves to this task.

Image similarity is something very subjective and relative to the world we live in and the needs we have. For example, twins may look alike for everyone except the family, birds look the same for a layman but a trained observer might recognize single individuals. It seems that the more we encounter an object or situation, the more detailed and differential description we have of it. Consequently, an image can mean different things to different people.

*Example: Image interpretation is personal*

A person wants to buy a dog, but doesn't know what breed of dog. He shows the system an image of a dog (see figure 1) and wants to get other images of dogs so that he can see what kinds there are. Another person shows the system the same image but only wants images of dogs from the same breed. A third person may only be interested in long-haired dogs etc.



Figure 1: Lassie

Furthermore, humans also tend to abstract query images for some conceptual information. We tend to associate objects in terms of our ability to interact with them (see [18]). This phenomenon can also be traced in text-based systems where the categories often represent actions (or corresponding nouns). For example, *glasses* can look very different from each other (see figure 2) but are still associated because we can perform a common action on them, namely *drink*.



Figure 2: Images of glasses

It is the authors belief that a truly useful system for general browsing has to be able to perform this association, but this is a very difficult task to accomplish in practice.



Figure 3: Rhyming images: a starfish and the statue of liberty

Today's CBIR-systems are not very capable to mimic human retrieval and need to be combined with traditional textual search. Text is designed (by evolution) to be a common interface between people and allows us to communicate with each other. Manual annotation of keywords to every image in a database is however a tedious work and, since the annotator is only human, he is bound to forget useful keywords. Also, keywords cannot capture abstract concepts and feelings like freedom and joy which have different meanings to each individual person, even though some attempts in this direction exist. The old saying 'An image says more than a thousand words' definitely still holds.

Often the result of a query in a CBIR system is not what the user expected. If for instance a user sketches or clicks on an image of a starfish because he wants to find more images of starfishes, he might get the Statue of Liberty in return simply because the system uses points as an image feature (see figure 3). This phenomenon is sometimes called *rhyming images*.

There are many things to consider in the design of a system for content-based search in image databases:

- Image features - What visual features are most useful in each particular case?
- Image representation - How should we code the image features?
- Representation storage and retrieval - The search must be made fast. What are the proper searching techniques and indexing structures?
- User interface - How should the user best browse and search for images? What is a suitable 'visual language'?

This survey will focus on image features and the user interfaces that are used in several systems and algorithms used for content-based search in image databases today. It contains both an overview and project-specific descriptions of these matters. The survey concentrates on search in still image databases even though some of the systems also support search in video data. Most of the systems in this report are prototype systems, available on the Web or elsewhere, while some are commercial products or algorithms.

This report is written as a HTML-document with hyperlinks to sites and demos from different projects. The later part of the survey (details and links) is well suited for the HTML-format because there are many on-line demos, publications and other facts on the Web. It is also easy to add new projects into the document.

Most of the images in specific project descriptions in this report were downloaded from homepages and other reports on respective project.

## 2 Applications

### 2.1 General

The applications is sometimes divided into three categories depending on the goal of the query:

- **Target search:** The target image is known (but the users memory of it might not be exactly correct, he will misplace some objects, ask for a slightly different color distribution etc.).
- **Category search:** Search images from a category (e.g. a person wants to buy a dress, but is not sure of exactly how it should look, she has only a notion of the general texture and tone of color).
- **General browsing:** The goal can be very vague or even unknown - "I know it when I see it".

The first category is an objective search, while the other two are subjective. There are many applications that fall into one of the three categories above:

- Digital libraries
- Face matching for identification and law enforcement
- Medical image analysis
- On-line shopping
- Trademark or logo searching
- Internet publishing and searching
- Fingerprint identification
- Remote sensing systems looking for abnormalities in satellite or aerial imagery
- Military uses like target identification

### 2.2 Examples

This section presents some real cases where content-based search is needed or already being used.

- **Alta Vista:** <http://image.altavista.com/cgi-bin/avncgi>  
- see Virage project.
- **Yahoo image surfer:** <http://isurf.yahoo.com/>  
- see Excalibur project.
- **State of California Department of Water Resources (DWR)**  
(<http://wwwdwr.water.ca.gov/>),  
Photography Unit.  
([http://wwwdwr.water.ca.gov/owegs/dir-Photo\\_Unit/photo.html](http://wwwdwr.water.ca.gov/owegs/dir-Photo_Unit/photo.html))

from Chabot, for more information see [40]:

The agency oversees the system of reservoirs, aqueducts and water pumping stations throughout California known as the State Water Project. DWR maintains a growing collection of over 500,000 photographs, negatives, and slides, primarily images of State Water Project facilities, but also many images of California natural resources.

Over the years, as DWR has made its collection available to the public, it has found itself devoting increasing resources toward filling requests for prints and slides. The agency receives 100-150 requests a month from a variety of sources: other government agencies, regional magazines, encyclopedia, university libraries, wildlife organizations, and individuals. Requests vary from those where the ID number of the desired picture is already known, to very general requests for "scenic pictures" of lakes and waterways.

The DWR application require a system that has to be flexible enough to handle complex queries that combine several of the attributes of the images, e.g. "Find a picture of a sunset taken near San Francisco during 1994".

- **United States Patent and Trademarks Office (USPTO)**

(<http://www.uspto.gov/>)

from Jain and Vailaya, for more information see [26]:

There are over one million registered trademarks in the U.S. alone, and they represent a number of goods and products which are sold by different manufacturers and service organizations. A trademark is either a word, phrase, symbol or design, or a combination of words, phrases, symbols or designs, which identifies and distinguishes the source of goods or services of one party from those of others. Most of the trademarks are abstract representations of concepts in the world, like abstract drawings of animals, or physical objects (sun, moon, etc.).

The trademarks are organized in an hierarchical tree structure. The highest level consists of 30 main categories (including celestial bodies, human beings, animals, geometric figures, foodstuffs, etc.) Each category is further divided into divisions (e.g. celestial bodies is divided into stars and comets, constellations and starry sky, sun etc.), which are further divided into sections (e.g. sun rising or setting, sun with rays or halo, sun representing a human face, etc.). A trademark can belong to more than one branch in the tree.

### 3 Some projects around the world

This section is divided into two subsections. The first one is an overview of the features and user interfaces that is used in content-based image search systems. It also contains a table with an overview of the systems and their features and user interfaces. The second one contains a bit more detailed descriptions on each project with links to demos and papers on the Web.

## 3.1 Overview

### 3.1.1 Image features

Useful features in general browsing is often considered to be the ones that mimic features in the human vision system. Using these increases the chance that the CBIR system retrieves images that looks similar from a human point of view. For example colors does not exist in reality, they are just a model of the light spectrum, but they are used in human vision and are therefore good feature candidates (as opposed to e.g. infrared light). Other examples are simple textures and edges. These are all low-level (primitive, preattentive) features and seems to be common in most peoples vision system (see e.g. [3]). No one really knows what goes on beyond these features in the human visual pathways. There are some indications that high curvature points (e.g. corners) might be one of the higher level features.

This is not to say that other features never should be used. There exists many applications where domain specific features is useful, e.g. MR and PET images for medical applications and infrared light for satellite images. Every application require a custom solution.

There are different ways to represent the image content:

- *Global features:*  
Includes feature average value, standard deviation, and histograms (probability density functions) calculated globally for the entire image.
- *Local features:*  
The features can be used locally for template matching or each local feature can be accompanied with position.
- *Object features:*  
Features can be calculated for each object separately. Objects can be segmented manually or by several automatic and semi-automatic ways:
  - Simple thresholding if the object is easily distinguished, e.g. birds against the blue sky.
  - Flood-fill technique: The user starts by marking a pixel inside the object and adjacent pixels with sufficiently similar color are included (a dynamic threshold can be selected by having the user click on both background and object points), see project QBIC.
  - Elastic shape model: (also called active contours, snake, and deformable templates): An elastic shape model (c.f. rubber-band) approximating the contour of the object (manually automatically selected) is automatically aligned to the object using feature points on the boundary, e.g. edges and high curvature points. The model can be to be more or less stiff depending on how detailed description of the object the user desire.  
This will give a representation of the boundary contour, also referred to as *chain-code*. For more details, see e.g. projects Photobook and QBIC.
- *Region features:*  
The image can also be divided into regions having homogeneous properties in some sense (e.g. color regions). The regions can also be fix regions in-

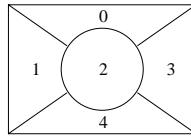


Figure 4: Example of regions from project PicSOM

dependent of image content, e.g. blocks or other structures, see for example figure 4.

The border between objects and regions is somewhat fuzzy. Regions from homogeneous properties can sometimes be interpreted as objects.

The features are typically calculated off-line and stored for each image so efficient computation is not a critical criteria (as opposed to efficient image/feature *matching* in the query stage), but if the database consists of many hundred thousand images this step can take a while.

They are sometimes divided into *low-level* features and *high-level* features. The high-level features are often calculated from the low-level features. Some of them are not as general as the low-level features, but are instead domain specific in the sense that they are designed to detect a special complex class of images or objects, e.g. faces, trees etc, which require training using manually selected representative images. The high-level features are therefore divided into *general* and *domain specific* features.

#### Low-level features:

- *Color:*

Includes gray-scale intensity and color extracted from different (in some sense natural) color spaces, e.g the

- *RGB (Red, Green, Blue) space*

- *YIQ and YUV spaces:*

Are used by the television standard committees PAL (Europe) and NTSC (America) as composite color standard. This color space is a linear transformation of the RGB space. The Y component contains the luminance and is the one shown in black-and-white televisions (calculated as  $Y = 0.30R + 0.59G + 0.11B$ ). The chromaticity is encoded in the other components.

- *HSV (Hue, Saturation, Value or Brightness) and HSI (Hue, Saturation, Intensity) space*

This color space is closer to human color perception than RGB and YIQ but not a perfect model, for instance two different colors of the  $I = 0.5$  plane are not all of the same perceived brightness. The HSI space is sometimes called the HLS (Hue, Lightness, Saturation) space.

Figure 5 contains the RGB, HSV and HSI color spaces.

- *Texture:*

Includes almost any local simple features such as energy, entropy, homogeneity, coarseness, contrast, correlation, cluster tendency, anisotropy, phase,

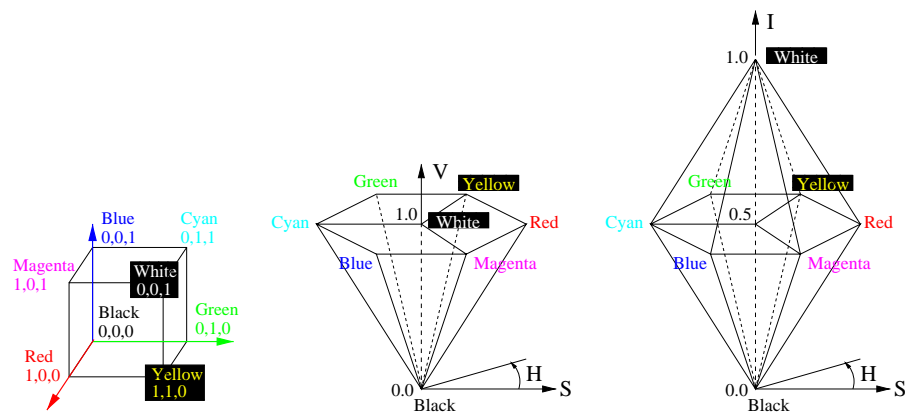


Figure 5: RGB, HSV and HSI color spaces



Figure 6: Some examples of texture from the Brodatz database (from Photobook)

roughness, directionality, flames, stripes, repetitiveness, granularity, etc. For some examples of textures, see figure 6.

Textures are detected with a number of different techniques, e.g. Tamura model, Markov random fields and Wold model.

- *Edges/Local orientation:*

Edges and their directions (local orientations) can be extracted in several ways, e.g. by using Gabor filters, Sobel operators, Canny edge detector, image gradient or Laplace filters.

- *Wavelet transform:*

Several types of wavelet transforms has been used, e.g. Haar (fast and simple to implement and the block-effect problem in image coding is not an issue here), Daubechie, Gabor. They detect simple features in several scales.

There is not a clear distinction between the above categories. Wavelets can be interpreted as edge information and edges is sometimes referred to as a texture measure.

**High-level features:**

**(General):**

- *Curvature (Shape):*

The most used curvature measure  $\kappa$  is the rate of change in tangent direction  $\phi$  of the object or region boundary contour, as a function of arc length  $u$ , see figure 7. If the object were a circle with radius  $R$  we would get  $\kappa = 1/R$ .

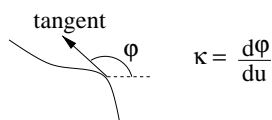


Figure 7: Curvature

It can be shown that the curvature can be calculated from the contour  $(x(u), y(u))$  as

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{(\dot{x}^2(u) + \dot{y}^2(u))^{3/2}}$$

where  $\dot{\phantom{x}}$  and  $\ddot{\phantom{x}}$  means the first and second derivative with respect to arc length  $u$ . The curvature does not have to be calculated from the boundary contour description (chain-code), it can also be calculated in any local region of the image, e.g. using polynomial fitting to model the image gradient and higher derivatives (this can be implemented by linear filters).

- *Shape:*

Regions and objects can be represented with its boundary contour (chain-code). From this a set of object shape descriptors can be calculated such as:

- Simple Shape features: area, circularity (perimeter<sup>2</sup>/area), eccentricity (major axis/minor axis), major axis orientation, longer/shorter axis.
- Shape complexity: For example
  - \* Fourier descriptors: DFT of the boundary contour described as complex numbers  $x(u) + iy(u)$ . Certain simple manipulations of the Fourier coefficients can eliminate dependency of position, size, and orientation, e.g. the description can be made rotation invariant by taking the magnitude of the Fourier coefficients.
  - \* Shape moments
  - \* image shape spectrum: Histogram of curvature.

Another method is to use the contour more directly in the object matching process. One way to compare two objects is to compute the amount of energy needed to align or deform the shape model of one object to the shape model of the other object. If the energy required to align the two models is relatively small, then the objects are very similar, see e.g. projects Photobook and Jain and Vailaya.

- *Position:*

Spatial location of features, regions (e.g. color areas) or objects.

- *Spatial relations:*

Relative locations between objects or regions (e.g. 'object 1 is located above and to the left of object 2').

**(Domain specific):**

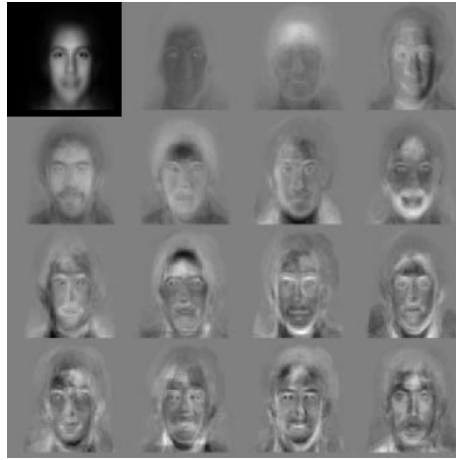


Figure 8: Eigenfaces, example from Photobook

- *Eigenimages:*

Karhunen-Loeve transform (also referred to as Hotelling transform) uses principal component analysis to find uncorrelated prototype images that best describes a class of images.

One example is *eigenfaces* for face recognition, see figure 8. From a set of training images containing faces,  $\{\mathbf{x}_k\}$ , one can form the covariance matrix  $\mathbf{C} = E\{(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T\}$  where  $\mathbf{m} = E\{\mathbf{x}\}$  (and the images  $\mathbf{x}_k$  is reshape into vectors). Let  $\mathbf{v}_n$  and  $\lambda_n$  be the eigenvectors and the eigenvalues to  $\mathbf{C}$  (we can write  $\mathbf{C} = \sum_n \lambda_n \mathbf{v}_n \mathbf{v}_n^T$ ). The eigenvectors with the largest eigenvalues contains most information in some sense and can be thought of as prototype images (in the case of faces, they are called eigenfaces). Each image in the set can be approximated with a linear combination of these eigenvectors,  $\mathbf{x}_k \approx \sum_p c_p \mathbf{v}_p$ . The coefficients  $c_p$  is the feature description for the image  $\mathbf{x}_k$ .

- *Learned feature histograms:*

From each class of images a set of features is calculated. Each class of images then has a set of feature vectors for which a histogram is calculated, which is a model of the probability density function for the class feature vectors. Each class is represented by its histogram. An example of this can be found in project ALISA.

- *Manually marked interesting areas:*

The user can mark interesting areas, not necessarily objects, from which further features is calculated. An example is a physician marking interesting areas in the lungs, see project ASSERT.

- *Templates (Shape):*

In the simplest case template matching means cutting out a region containing the object and use this as a filter to correlate with the images in the database to find the same object.

The match can be made invariant to rotation and scale if several rotated and scaled versions of the filter is used. Scales, rotations.

- *Attributed relational Graphs (Shape):*

Objects and images can be represented with graphs with nodes and relational attributes between nodes, one example of this is project Huet and Hancock. The nodes can for example contain line segments. Attributes between nodes can then be calculated such as the angle and normalized distance (a distance measure invariant to scaling) between node line segments. The similarity between a node in one graph to a node in another graph can be made by comparing their relational attributes to the other nodes in their respective graph.

If the attributes is invariant to translation, orientation and scale, then so is the object matching.

The matching procedure is complex and can be time consuming (compare many nodes in one graph with many nodes in another graph using the node relations to find best matching pairs of nodes and use these to calculate similarity between two objects).

- *Text:*

Keyword annotation or division into categories and sub-categories. They are usually manually annotated, but they can also be labeled by *power-assisted annotation* - the user labels a few images and the system finds similar images using the image features and label those images by itself, see Photobook.

It can be mentioned that for video data some extra features are used:

**Video features:**

- *Motion*

- *Scene breaks or cuts:*

The most important frames in the video sequence is often considered to be the ones that is located in the beginning and the end of a scene.

- *Editing features:*

E.g. dissolves, fades, wipes, zooming

Some of the low-level features above corresponds to *preattentive* features in human vision. They are considered to be computed in parallel without conscious involvement as opposed to *attentive* features which are serially computed and require conscious guidance.

Matching the more complex features in CBIR systems is often computationally expensive and the preattentive features are therefore sometimes used in a pruning stage to remove most of the non-relevant images in the database before matching the complex features in the remaining set.

The features is usually composed into a multidimensional *feature vector*. If the features are properly chosen each class of images form *clusters* in the feature space, see figure 9 for an example.

The similarity metric is usually defined via a distance measure which will be used for nearest neighbor match in the feature space. Examples of distance metrics is the Minkowski distance  $(\sum_k |a_k - b_k|^r)^{1/r}$  (e.g.  $r = 1$  gives  $L_1$  and  $r = 2$  gives  $L_2$ ), weighted least square, or city-block distance. The distance metric can also be

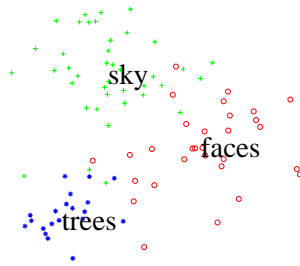


Figure 9: (2D) Feature vectors building clusters

based on a statistical approach by assuming a model for the probability function and use maximum likelihood.

The similarity measure can be fix or adapted by user choice, which in turn can be done by letting the user choose the feature weights or by relevance feedback (see below).

The query should be fast, preferably in the order of a few seconds. There are several attempts to speed up the search, for example:

- *Reduce feature space dimension:*  
Some features are probably more useful than others. The dimension and therefore the complexity of the feature space can be reduced if only the most useful features is used. One example is to use Karhunen-Loeve transform to find the best linear combinations of features, see e.g. project LCPD, Leiden.
- *Vector quantization:*  
Several different techniques exists which aim to quantize the feature space. For instance a prototype vector can be used to represent each cluster. The search can then be made by first comparing the query feature vector with the prototype vectors to find the closest cluster and second to find the closest feature vectors within that cluster.  
The quantification can also be calculated in a hierarchical manner. Two examples are project Multiresolution search, which uses tree-structured vector quantization (TSVQ) and project PicSOM, which uses Tree-Structured Self-Organizing Maps (TS-SOMs).

### 3.1.2 User interface - visual language

The choice of user interface or visual language is an very important step in the design of a CBIR system.

In the systems today query is made by one of the following methods:

- *Example image:*  
Simply show the system an image with an example and retrieve similar images.
- *Selected features (feature weights):*

The user selects features (including similarity metrics) he thinks might characterize the type of images he is looking for, e.g. use color and texture features to find dresses or wallpaper. The selection can for instance be made by weighting the available features according to their relevance.

- *Example image + selected features or regions:*

A combination of the two above. Here the user selects an image and tells the system what is important the image by selecting features or regions/objects (the regions/objects can either be automatically or manually segmented).

- *Sketch:*

Instead of showing an example image, the user can sketch his own image. The sketch can e.g. be a contour sketch showing the shape of the objects to retrieve or in form of colored boxes indicating color, size and position of the objects.

Sketch = description and not necessarily a hand drawn contour.

- *Image icons:*

A variant of the sketch approach where the user creates an example image by selecting predefined icons of e.g. human faces, trees, sky etc.

One advanced example used for search in video data is project Media streams (see [11]). It uses 3500 icons, which are sorted and grouped hierarchically in descriptive categories. They are structured to deal with the special semantic and syntactic properties of video data, including: space, time, weather, characters, objects, character actions, object actions, relative position, screen position, recording medium, cinematography, shot transitions, and subjective thoughts about the material.

- *Relevance feedback:*

The user grades the retrieved images by their relevance, e.g. highly relevant, relevant, no-opinion, non-relevant, or highly non-relevant. The system then uses the grades to learn relevant features and similarity metrics. The grading can be made after each new retrieval, thereby iteratively improving the result.

In the general case, we cannot expect to find a similarity measure that suites all people and needs. A user should therefore be given the possibility to interact with the system and the similarity be learned by the system. This interaction can be more or less natural for the user. For instance, manually selecting relevant features can be difficult, especially when the features are complex.

Some systems, e.g. blobworld, allows the user access to the systems internal representation of the image. This can help to understand how the system reasons and ease the selection of proper features.

Relevance feedback goes a step further and removes the burden of manually specifying the feature weights. This approach is getting increasing attention and may turn out to be the most promising one.

### 3.1.3 Evaluation methods

The evaluation of the performance of an algorithm or system is a difficult problem. The systems are usually evaluated using one of the following methods:

- *Test-images from a certain category:*

Many algorithms and features are developed to carry out a certain domain specific task, e.g. categorize texture images, medical images, satellite data, face images, some 3D objects from different angles, or marine animals. They are thus not designed for general browsing and it might be easier to define a goal.

- *Test-Images with 'ground-truth' (keywords or categories):*

For general browsing systems, the evaluation is somewhat harder. Some large databases have their images sorted into a few categories (e.g. animals, people, art, etc.) and sub-categories (e.g. dog, cat, horse, etc. for the animal category). Some databases even have several keywords attached to each image. These labels can be used as a simple kind of ground-truth. But, as said before in this survey, keywords is not always sufficient information so they should not be interpreted as a true ground-truth.

A few databases have been used by more than one project. This allows systems and algorithms to be compared against each other, which can serve as an evaluation method. The best example of such a database is the Corel photo database <http://www.corel.com/>. It contains many thousands of images with attached keywords. Corel photos has been used by e.g. projects Blobworld, MARS, NETRA, PicSOM, and SIMPLIcity.

- *Test-images without ground-truth, subjective evaluation:*

Browsing systems using Web images does not have a ground-truth, except keywords from the images filenames and URL:s. The evaluation is mainly subjective by simply make some queries and look at the result.

### 3.1.4 Projects overview

The table below contains an overview of some of the CBIR projects that exists today. For each project is marked the image features and the interface used and also the status of the project.

The abbreviations in the feature columns (G,L,O,R) refers to the image representation rather than how the features are extracted. The features can be calculated directly from the image or from other features.

One must also bare in mind that each feature category include a wide variety of features and they can sometimes overlap.

G = global feature, L = Local feature, O = Object feature, R = Region feature  
 E = Example, F = selected Features or Feature weights, RF = Relevance Feedback,  
 S = Sketch  
 A = Algorithm, P = Prototype, C = Commercial

Project	Source	Features										Query by	Status	
		Color	Texture	Shape	Pos.	Spatial Rel.	Edges	Text	Other					
ALISA	Washington Univ. + companies	L	L	L			L				L		E	C
ASSERT	Purdue Univ. + others		R	R			R				R		E, RF	P
Blobworld	UC Berkeley	R	R		L	R						•	E+F	P
CANDID	Los Alamos Nat. Lab.	G	G				G						E	P
Chabot	UC Berkeley	G	G									•	F	P
Excalibur	Excalibur Technologies	G	G	G			G				G	•	E+F	C
Fast Multiresolution	Washington Univ.	G+L									(L)		E	P
ImageSearch	Leiden	O	O	O	O		O				O		S	P
LCPD, Leiden	Leiden Univ., Netherlands	(L)	G		L		L				L		E+F, RF	P
MARS	Univ. of Illinois	G/(R)	G/(R)	O			(L)				(L)		RF	P
Multiresolution search	Purdue Univ.	(R)	(R)				(R)				(R)		E+F	P
NETRA	UC Santa Barbara	R	R	R	R		R				R		E+F	P
PhotoBook/FourEyes	MIT Media lab		G	O			O					•	RF	P
PicSOM	CIS, Helsinki Univ.	(R)	(R)	(R)			(R)				(R)		RF	P
QBIC	IBM, Almaden Research Center	G/O	G/O	O			O				O	•	E, F, S	C
SIMPLICity	Stanford Univ.	R	R	R	R		R						E	P
SQUID	CVSSP, Surrey Univ.			O									E	P
Surfimage	IMEDIA, INRIA, France	G	G	G			G				G		(E+F), RF	P
Virage	Virage Technologies, Inc.	G, L	G	O			O					•	E+F	C
VisualSEEK/SaFe	Advent group, Columbia Univ.	G, R		R	R	R	R	R	R	R	R		S, E	P
WebSEEk	Advent group, Columbia Univ.	G										•	E, F, RF	P
Huet and Hancock	Univ. of York			G/O			L						E	A
Jain and Vailaya	Michigan State Univ.			G			G						E	A

<sup>a</sup>Adaptive features, learned by the system

<sup>b</sup>Manually marked interesting areas

<sup>c</sup>Special features for fingerprints and faces

<sup>d</sup>Haar wavelet decomposition

<sup>e</sup>Wavelet decomposition

<sup>f</sup>Appearance-specific description (e.g. eigenfaces)

<sup>g</sup>Eigenimages, Flexible images, Image shape spectrum, Wavelet transform

<sup>h</sup>Domain specific, e.g. eigenfaces for face recognition

<sup>i</sup>QMF Wavelet transform

## 3.2 Details and links

### 3.2.1 ALISA (Adaptive Learning Image and Signal Analysis)

Source	<p><a href="http://www.seas.gwu.edu/~pbock/ALISA_Summary.html">http://www.seas.gwu.edu/~pbock/ALISA_Summary.html</a> and the ALIAS Corporation at <a href="http://hudson.idt.net/~jrhals/alias.html">http://hudson.idt.net/~jrhals/alias.html</a></p> <p>Peter Bock and others at George Washington University and people at the Research Institute for Applied Knowledge Processing (FAW) in Ulm, Germany.</p>
Features	<p>ALISA uses a learning algorithm called Collective Learning developed by Bock to build a statistical representation of an image. This representation can then be used to compare images or image regions. ALISA can look for characteristics of entire images or parts of images. Collective Learning has been shown to learn much faster than other popular adaptive systems like neural networks. ALISA can be trained to recognize a new texture or shape with only a few images. ALISA consists of several layers:</p> <ol style="list-style-type: none"><li>1. Texture module: Receives input from the environment via a set of feature sensors, e.g. average pixel intensity, standard deviation, skew, kurtosis, gradient magnitude, gradient direction, hue, saturation, and brightness. Feature parameters usually include the size of the analysis token, the dynamic range of the feature value, and the precision of the feature value. The parameters is often manually selected, but can also be automatically selected. A feature vector is calculated in each pixel of every image. Each class of images then has a set of feature vectors for which a histogram is calculated (typically 10 to 100 images is sufficient to estimate the histogram). Each class is represented by its histogram. To classify a pixel (or local area) a feature vector is again computed. The histogram which has the highest posterior probability for that feature vector is then selected (e.g. each pixel in an image is classified, not the whole image). If only one class is learned, this is called <i>anomaly detection</i>. The pixel is classified as an anomaly if the probability of the feature vector belonging to the histogram is above a certain threshold.</li><li>2. Geometric module: Uses the information in the texture module to classify simple geometric concepts like boundaries of regions. These concepts are divided into <i>canonical concepts</i> (non-cultural, e.g. horizontal, curved, symmetric, slanted, intersected) and <i>secular concepts</i> (cultural, e.g. wavy, cratered, polka-dotted, text and graphic structures). The Geometry Module is trained in the same manner as the texture module except that it uses a fixed 3x3 token to profile the directionality of edges and edge intersections in an image.</li><li>3. Shape module: Uses the information from the geometry module to classify complex configurations of edges and regions. The Shape Module uses a scale and rotation invariant token to find characteristic relationships among neighboring edges.</li><li>4. Concept module: Combines the decisions represented in the other modules into a summary classification.</li></ol> <p>The modules can be alternated and used in several layers. The shape module is under development and the concept module is future research.</p>

*continued on next page*

<i>continued from previous page</i>	
Interface	<p>ALISA operates in three phases:</p> <ol style="list-style-type: none"> <li>1. Training: ALISA is shown a short number of representative images for each class.</li> <li>2. Control: The system is shown a representative image (not used in the training phase) for each class. If the system cannot classify the image correctly, the system needs further training or feature refinement.</li> <li>3. Test: The learning is disabled and the system can be used for classification.</li> </ol>
Publications	<a href="http://www.seas.gwu.edu/~pbock/">http://www.seas.gwu.edu/~pbock/</a> , e.g. [6]
Test-Images	<p>Successful applications for the ALISA texture and geometry modules include</p> <ul style="list-style-type: none"> <li>• Detection of vehicles in desert terrains</li> <li>• Detection of intruders in restricted environments</li> <li>• Detection of aneurysms in MRI heart images</li> <li>• Identification of defective motors from their acoustical signatures</li> <li>• Classification of malignant cells in liver samples and mammograms</li> <li>• Classification of batteries by brand, coins, fruits, fonts, etc.</li> </ul>
Comments	ALISA has been used in industrial applications since 1990 and is commercially available from Robert Bosch GmbH in Germany and the ALIAS Corporation in North America.

### 3.2.2 ASSERT (Automatic Search and Selection Engine with Retrieval Tools)

Source	<p><a href="http://rvl2.ecn.purdue.edu/~cbirdev/WWW/CBIRmain.html">http://rvl2.ecn.purdue.edu/~cbirdev/WWW/CBIRmain.html</a></p> <p>Chi-Ren Shyu, Carla Brodley, Avi Kak and Akio Kosaka, Purdue University, School of Electrical and Computer Engineering          Alex M. Aisen, Indiana University Medical Center          Lynn S. Broderic University of Wisconsin Hospital</p>
Features	<p>The features are extracted as follows:</p> <ol style="list-style-type: none"> <li>1. A human (physician) marks interesting areas called PBR (Pathology Bearing Landmarks).</li> <li>2. Automatically extract the lung region using binary-image analysis routines.</li> <li>3. Extract local features from each PBR and global features from the entire lung region. There are 255 features extracted, e.g             <ul style="list-style-type: none"> <li>• <b>Texture:</b> Using a statistical approach based on gray-level co-occurrence matrix a set of texture features are extracted such as <i>energy, entropy, homogeneity, contrast, correlation</i> and <i>cluster tendency</i>.</li> <li>• <b>Gray-scale properties:</b> Mean and standard deviation of the gray-levels with respect of the pixels in the rest of the lung and a histogram of the local gray-levels.</li> <li>• <b>Shape:</b> Longer axis, shorter axis, orientation, shape complexity measurement using both Fourier descriptors and moments.</li> <li>• <b>Edges:</b> Using Sobel edge operator to compute edge distribution (ratio of the number of edge pixels to the total number of pixels in the region for different edge threshold channels)</li> <li>• <b>Structure of gray-level variations within the PBR:</b> Compute the number of segmented regions in a PBR, histogram of the sizes for all these regions, and a gray-level statistics for each region.</li> <li>• Distance between the centroid of a marked PBR and the nearest lung boundary.</li> <li>• Position of PBRs in the lung.</li> </ul> </li> </ol> <p>The dimension of the set of feature vectors is reduced into 12 attributes by applying sequential forward selection search (SFFS) which finds the most useful features.</p> <p>A multi-dimensional hash table is used to index the features efficiently. The main idea is to build a decision tree based on the minimization of average entropy. The 12 attributes selected from the SFFS algorithm will be used to build the decision tree first and the remaining will be used if they still have the power to minimize the average entropy.</p>
<i>continued on next page</i>	

*continued from previous page*

**Interface** The system is implemented in a human-in-the-loop approach: To archive an image, a physician delineates PBR (Pathology Bearing Landmarks) and anatomical landmarks. The system then creates a feature vector that characterizes the image. The physician can, for each of the retrieved images, elicit feedback to the system. If the physician disagrees with the system, that information can be used to alter the weights of the factors in the similarity metric.



Left top: Query image. Bottom row: Four best retrieved. Right top: Enlarged view of retrieved image. Right column: Adjacent images to the retrieved image.

**Publications** <http://RVL.www.ecn.purdue.edu/RVL/CBIR/CBIRPublications.html>, e.g. [50], [49]

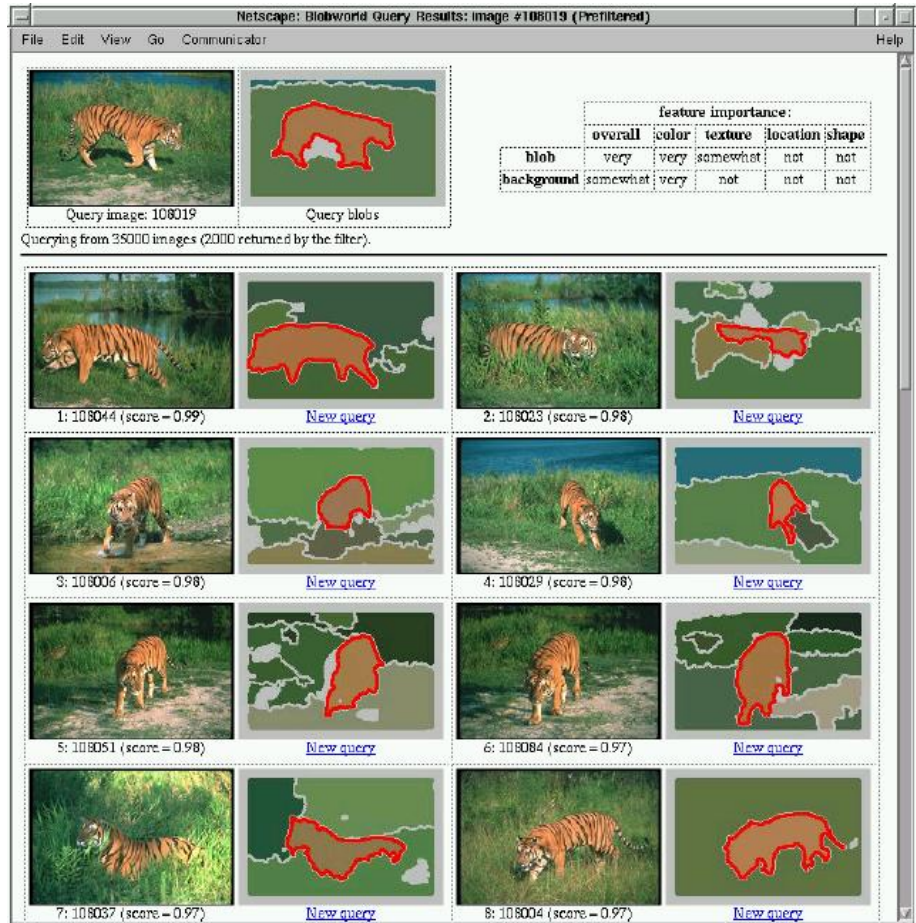
**Test-Images** HRCT (High Resolution Computed Tomography) of healthy and diseased lung images. About 300 images from 80 patients.

### 3.2.3 Blobworld

Source	<a href="http://galaxy.cs.berkeley.edu/photos/blobworld/">http://galaxy.cs.berkeley.edu/photos/blobworld/</a> UC Berkeley Computer Vision Group
Features	<p>Color and texture of segmented image regions (blobs). They are calculated as follows:</p> <ol style="list-style-type: none"> <li>1. Extract for each pixel           <ul style="list-style-type: none"> <li>• Color: three-dimensional <math>L^*a^*b^*</math> color space. The color vector is extracted from a smoothed version of the image.</li> <li>• Texture: three-dimensional space with the components <i>polarity</i> (quadrature phase), <i>anisotropy</i>, and <i>normalized texture contrast</i>. The last two is extracted from an orientation tensor (c.f. Harris (Plessey) corner detector) defined as <math>M(x, y) = g(x, y) * (\nabla I)(\nabla I)^T</math>, where <math>g</math> is a gaussian low-pass filter and <math>\nabla I</math> is the image gradient. The anisotropy is then defined as <math>1 - \lambda_2/\lambda_1</math> and the contrast as <math>2\sqrt{\lambda_1^2 + \lambda_2^2}</math>, where <math>\lambda_1, \lambda_2</math> are the eigenvalues of <math>M</math>.</li> <li>• Position</li> </ul> <p>in an appropriate scale (automatically selected from consistency in polarity scale-space).</p> </li> <li>2. Group the 8-dimensional feature vectors into regions by modeling the feature vector distribution with a mixture of Gaussians using an EM-algorithm (Expectation-Maximization). Use some post-processing to improve the segmentation.</li> <li>3. For each segmented region, use color histogram, mean texture contrast and anisotropy as feature descriptors.</li> </ol>
<i>continued on next page</i>	

continued from previous page

Interface The user can see the blob image, choose the interesting blobs, and finally specify the relative importance of the blob features. The user can specify disjunctions and conjunctions, e.g. "like-blob-1 and like-blob-2" and spatial relationship between two blobs, e.g. "like-blob-1-left-of-blob-2". The user can also select one blob and its complement, the 'background'. The compound query score is calculated using fuzzy-logic operations. Because the user has access to the internal representation (the blob image), it is easier to understand and choose appropriate feature weights.



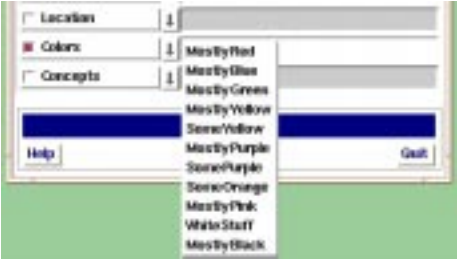

Publications <http://www.cs.berkeley.edu/~carson/research/publications.html> e.g. [8] and [5]

Test-Images Corel Stock Photos (<http://www.corel.com>)

### 3.2.4 CANDID (Comparison Algorithm for Navigating Digital Image Database)

Source	<a href="http://public.lanl.gov/kelly/CANDID/index.shtml">http://public.lanl.gov/kelly/CANDID/index.shtml</a> Los Alamos National Laboratory (Operated by the University of California for the US Department of Energy)
Features	<p>First a number of local features is extracted, e.g. color and/or texture, at every pixel location. Instead of making a histogram with a discrete number of bins of the feature vectors, a continuous probability density function (pdf) is calculated over the multidimensional feature space. This probability density function is the (global) signature of the image.</p> <p>The pdf is estimated as a weighted sum of Gaussian functions (a k-means clustering algorithm followed by an optional cluster merging process is used) (thus the signature for the image is really a set of mean vectors and covariance matrices). Several similarity measures between two probability density functions is proposed, e.g. the <math>L_2</math> distance measure or the angle (correlation) between the functions.</p> <p>There is also a suggestion that it would be a good idea to subtract the "average" of all signatures in the database as an attempt to get rid of the "background" before comparing signatures.</p>
Interface	Simply click on an image and get similar images.
Publications	<a href="http://www.c3.lanl.gov/~kelly/CANDID/pubs.shtml">http://www.c3.lanl.gov/~kelly/CANDID/pubs.shtml</a> , e.g. [28], [27]
Test-Images	<ul style="list-style-type: none"> <li>• Multi-spectral satellite data (Landsat TM data) (6-dimensional color vector: R, G, B and 3 IR bands)</li> <li>• Pulmonary CT Data (4-dimensional Laws texture energy vector: Edge, Spot, Wave, and Ripple)</li> <li>• Smithsonian Image Database (3-dimensional color vector)</li> </ul>

### 3.2.5 Chabot

Source	<p><a href="http://http.cs.berkeley.edu/~ginger/aboutChabot.html">http://http.cs.berkeley.edu/~ginger/aboutChabot.html</a></p> <p>Virginia E. Ogle and Michael Stonebraker at University of California, Berkeley. Chabot is part of the Digital Library project at UC Berkeley.</p>
Features	<p>The system uses a relational database management system called POSTGRES, which provide search by a combination of text and color (called a <i>concept query</i>).</p> <p>The color histogram is computed using the Floyd-Steinberg quantization. Only 20 elements is used in the histogram.</p>
Interface	<p>The underlying mechanism to perform query by color is called <i>MeetsCriteria</i>. It takes two arguments: a color criterion such as "Some Orange" and a color histogram. The user selects a color criterion from a menu. This criterion is then incorporated into the query using the selected color histogram. For example, a picture of a field of purple flowers having tiny yellow centers qualifies as "Mostly Purple", but you can also retrieve this picture using the search criterion "Some Yellow".</p> <p>In addition, the user can define his very own menu buttons for contextual information such as "sunset" or "snow" by combining text and color.</p> <div style="display: flex; justify-content: space-around;">   </div>
Publications	<p><a href="http://http.cs.berkeley.edu/~ginger/">http://http.cs.berkeley.edu/~ginger/</a>, e.g. [40]</p>
Test-Images	<p>Images from California Department of Water Resources (DWR) (see sec 2.2), primarily images of State Water Project facilities, but also many images of California natural resources.</p>
Comments	<p>Chabot has now been renamed into a Web-accessible database called Cypress (which appear to only support text-based search) , see (<a href="http://elib.cs.berkeley.edu/photos/dwr/">http://elib.cs.berkeley.edu/photos/dwr/</a>).</p>


### 3.2.6 Excalibur Visual RetrievalWare

Source	<p><a href="http://www.excalib.com">http://www.excalib.com</a></p> <p>Excalibur Technologies</p>
Features	<p>The system creates a feature vector for the image based on</p> <ul style="list-style-type: none"> <li>• <b>Color:</b> Including their relative proportions. The position of the colors does not matter.</li> <li>• <b>Shape:</b> Relative orientation, curvature, and contrast of lines in the image. The color, position and absolute orientation of the lines doesn't matter. (This attribute is most effective when the lines are crisp and clean.)</li> <li>• <b>Texture:</b> Flow and roughness on a small scale. The color, position, and orientation of these features doesn't matter.</li> <li>• <b>Brightness:</b> A measure of the brightness at each point in the image.</li> <li>• <b>Color structure:</b> A measure of the hue, saturation, and brightness at each point in the image.</li> <li>• <b>Aspect ration:</b> A measure of the ratio of the images width to its height.</li> </ul> <p>Special features are also supported that can match fingerprints and faces. The indexing technique is based on the Adaptive Pattern Recognition Processing (APRP) technology, which is developed by the founder of Excalibur, James Dowe. Based on neural network methods it acts as a self-organizing system that automatically indexes the binary patterns in digital information, creating a pattern-based memory that is optimized for the native context of the data.</p> <p>There is also a text retrieval system that uses a semantic network to link words so that a query can return not only exact matches but also close matches.</p>
Interface	Choose feature weights and click on an image to get similar images.
Publications	Information (not very detailed) can be found on the system homepage (link above) (e.g. [14]), [16] and [7]
Test-Images	Excalibur has developed demonstrations for fingerprint, face, and character recognition. The internet demo uses images from royalty-free clip-art CDROMs.
Comments	Among one of the first systems. Used by Yahoo Image Surfer, <a href="http://isurf.yahoo.com/">http://isurf.yahoo.com/</a>

### 3.2.7 Fast Multiresolution

Source	<p><a href="http://www.cs.washington.edu/research/graphics/projects/query/">http://www.cs.washington.edu/research/graphics/projects/query/</a></p> <p>Charles E. Jacobs, Adam Finkelstein and David H. Salesin at the University of Washington.</p>
Features	<p>The wavelet components is computed as follows:</p> <ol style="list-style-type: none"> <li>1. <b>Wavelet decomposition:</b> Perform a standard two-dimensional Haar wavelet decomposition (Haar because they are fast and simple to implement) for each color channel of the image. YIQ color space was used.</li> <li>2. <b>Truncation:</b> Throw away all but the <math>m</math> largest coefficients in each color channel to get a <i>sparse representation</i>. This both accelerates the search and reduces storage (and it also seems to improve the discrimination power).</li> <li>3. <b>Quantization:</b> Use only the sign of the <math>m</math> largest coefficients. The mere presence or absence of a feature appears to be efficient.</li> </ol> <p>The overall average color and the indices and signs of the largest <math>m</math> coefficients in each color channel is used as image 'signature'. This fast and easy to compute 'signature' gives encoded edge and color information.</p> <p>This metric essentially compares how many significant wavelet coefficients the query has in common with potential targets.</p> <p>The complexity of the algorithm is linear in the number of database images, and due to the fast computation of similarity using the defined metric, the overall running times are very short. Moreover, it is quite easy and fast to add new images to the database.</p>
Interface	<p>The input is a sketched or scanned image, intended to be an approximation of the image to be retrieved.</p> <p>Because the retrieval time is so fast (under 0.5 second in a database of 20000 images), the interface can have an interactive mode, in which the 20 top-ranked target images are updated whenever the user pauses for a half-second or more. In this case the user may not have to finish his sketch before he has found the target image.</p>
Publications	<a href="http://www.cs.washington.edu/homes/salesin/abstracts.html">http://www.cs.washington.edu/homes/salesin/abstracts.html</a> , [25]
Test-Images	Databases containing about 1000 and 20000 images.

### 3.2.8 ImageSearch (ImageScape)

Source	<p>http://ind134b.wi.leidenuniv.nl:2001/imagesearch.html</p> <p>Michael S. Lew, Kim Lempinen and Nies Huijsmans at Leiden University, the Netherlands</p>
Features	<ul style="list-style-type: none"> <li>• <b>Object:</b> Color, gradient, Laplacian, and texture information from every pixel of multiple scales is extracted. The Kullback relative information measure (related to Shannon's mutual information) is used to find the most informative 256 features for template matching which will minimize the misdetection rate.</li> <li>• <b>Shape:</b> The user can specify the contour of an object by sketching. The Sobel operator in conjunction with a Gaussian blurring filter is then used to find the edge/contour maps.</li> <li>• <b>Spatial position:</b> The algorithm return images which have roughly the same placement of objects. The search is made somewhat translational invariant by using low image resolution.</li> </ul>
Interface	<p>The user can draw a sketch or place representative image icons which refer to objects such as human faces, sand, trees, water, and sky.</p> 
Publications	[33], [32]
Test-Images	Over 25 million images from the World Wide Web.

### 3.2.9 LCPD, Leiden

Source	<p><a href="http://ind156b.wi.leidenuniv.nl:2000">http://ind156b.wi.leidenuniv.nl:2000</a></p> <p>Leiden Imaging and Multimedia Group (LIM), Leiden University, the Netherlands, and Philips Research Labs Eindhoven.</p>
Features	<p>The primitive feature used is intensity (color), image gradient (edge information), or texture. These features is the basis for several different image representations:</p> <ul style="list-style-type: none"> <li>• <b>Projection method:</b> Compares the horizontal and vertical projections of the intensity image or the gradient image. An image of size <math>n \times m</math> pixels gives rise to a horizontal and vertical projection vector of length <math>n + m</math>.</li> <li>• <b>Template method:</b> Compares the intensity image or the gradient image pixel by pixel based on spatial location.</li> <li>• <b>Texture method:</b> LBP (Linear Binary Patterns) is used which is two-level <math>3 \times 3</math> patterns where the threshold is the center pixel. This gives <math>2^9 = 512</math> possible texture units. A histogram of these features is then used to represent the image.</li> <li>• <b>KLT (Karhunen-Loeve transform)</b> (Hotelling transform) or <b>Fisher's LDA (Linear Discriminant Analysis)</b> (called <i>optimal key methods</i>). These methods is used to reduce the dimension of the LBP feature vector to its most important KLT or LDA part. Fisher's LDA requires a noise model and is useful in e.g. the print-scanner problem, while KLT is used for the general image degradation when the noise is considered unknown.</li> </ul> <p>KLT and Fisher's LDA were shown to require only about 10% of the features relative to the texture, projection and template methods, while still achieving equivalent or better accuracy.</p>
Interface	<p>The user can choose between two interfaces: Either make a query using relevance feedback or select an image and a combination of primitive features (intensity, gradient, texture), image resolution (18, 37 or 75 dpi), and image representation (KLT, projection, etc.).</p>
Publications	<p><a href="http://www.wi.leidenuniv.nl/home/lim/lim.html">http://www.wi.leidenuniv.nl/home/lim/lim.html</a> and <a href="http://www.wi.leidenuniv.nl/~huijsman/">http://www.wi.leidenuniv.nl/~huijsman/</a>, e.g. [31], [24]</p>
Test-Images	<p>Many thousands of portrait images from Leiden 19th Century Portrait Database (LCPD). Some images are copies of each other, however, due to different storage conditions, the copies have varying kinds and different amounts of degradation (e.g. moisture damage, scratches and writings on the images).</p>

### 3.2.10 MARS (Multimedia Analysis and Retrieval System)

Source	Yong Rui, Thomas S. Huang, Sharad Mehrotra, Michael Ortega and others at University of Illinois at Urbana-Champaign
Features	<p>The MARS system is based on the relevance feedback idea. The system has access to a variety of features and similarity measures and learns the best ones for the particular query by letting the user grade retrieved images as <i>highly relevant</i>, <i>relevant</i>, <i>no-opinion</i>, <i>non-relevant</i>, or <i>highly non-relevant</i> (so the burden of specifying the feature weights is removed from the user).</p> <p>Several different features has been tested within the MARS project:</p> <ul style="list-style-type: none"> <li>• <b>Color:</b> Represented in the HSV color space. Calculates color histogram and color moments.</li> <li>• <b>Texture:</b> Uses Tamura textures and co-occurrence matrix in different directions to extract e.g. coarseness, contrast, directionality and inverse difference moments.</li> <li>• <b>Shape:</b> Fourier descriptor, chamfer descriptor</li> <li>• <b>Wavelet coefficients</b></li> </ul> <p>The features can be extracted globally or from <math>5 \times 5</math> sub-images. The information in each type of feature (e.g. color) is represented by a set of sub-features (e.g. color histogram, color moments, etc.). The sub-features and similarity measures are normalized (using Gaussian normalization) to have the same dynamic range. This will ensure equal emphasis of each sub-feature within a feature type and equal emphasis on each similarity value.</p>
Interface	(the links to the demos did not work when last checked at the beginning of the new millenium)
Publications	<a href="http://research.microsoft.com/users/yongrui/html/publication.html">http://research.microsoft.com/users/yongrui/html/publication.html</a> or <a href="http://www.ifp.uiuc.edu/">http://www.ifp.uiuc.edu/</a> , e.g. [48], [36], [46], [47], [42]
Test-Images	<p>Several collections have been used (with different sets of features):</p> <ul style="list-style-type: none"> <li>• Collection from the Fowler Museum of Cultural History at the University of California-Los Angeles. 286 ancient African and Peruvian artifacts (part of the Museum Educational Site Licensing Project (MESL))</li> <li>• over 70000 Corel photos (<a href="http://www.corel.com">http://www.corel.com</a>)</li> <li>• Texture images from MIT Media Lab (VisTex database) (<a href="ftp://whitechapel.media.mit.edu/pub/VisTex/">ftp://whitechapel.media.mit.edu/pub/VisTex/</a>)</li> </ul>
Comments	Users from various disciplines, such as Computer Vision, Art, Library Science, etc., as well as users from industry, have been invited to compare the retrieval performance between the relevance feedback approach in MARS and the <i>computer centric</i> approach where the user specify the relative feature weights himself. All the users rated the feedback approach much higher than the computer centric approach in terms of capturing their perception subjectivity and information needed.

### 3.2.11 Multiresolution Image Database Search

Source	<p>Search: <a href="http://min.ecn.purdue.edu/~jauyuen/webfind.html">http://min.ecn.purdue.edu/~jauyuen/webfind.html</a>  Browse: <a href="http://min.ecn.purdue.edu/~jauyuen/webbrowse/webbrowse.cgi">http://min.ecn.purdue.edu/~jauyuen/webbrowse/webbrowse.cgi</a></p> <p>Jau-Yuen Chen, Charles A. Bouman, Jan P. Allebach and others at Purdue University, West Lafayette, IN</p>
Features	<p>Histograms of color, texture and edge features are calculated for a number of regions in the image (maximal <math>4 \times 4</math> regions). The features are defined as</p> <ul style="list-style-type: none"> <li>• Color: The <math>L^*a^*b^*</math> space was used. The color histogram was smoothed so that small color shifts did not effect the match.</li> <li>• Texture: The texture feature is formed by histogramming the magnitude of the local image gradient for each color component.</li> <li>• Edge: Formed by thresholding the edge gradient and then computing the angle of the edge for all points that exceed the threshold.</li> </ul> <p>The total feature vector had in their experiment a length of about 200. To speed up the search a tree-structured vector quantization (TSVQ) is calculated and the search is made by a branch and bound technique on this tree. The distance measure includes both distance to cluster mean and maximal distance between mean and cluster members. The distance measure also increases from top-down in the tree, ensuring that the similarity cannot improve when the search is made top-down.</p> <p>While exact search is possible, a free parameter allows search accuracy to be reduced, thereby providing a substantially better speed-up versus accuracy tradeoff.</p>
Interface	<p>The user can choose Similarity Metric, Finest Resolution (number of regions), Resolution Weight, Color Weighting (lightness, red-green and blue-yellow) and Feature weighting (color, texture and edge). The user then selects an image and the system retrieves similar images.</p>
Publications	<p><a href="http://albrecht.ecn.purdue.edu/~bouman/publications/pub_database.html">http://albrecht.ecn.purdue.edu/~bouman/publications/pub_database.html</a> or  <a href="http://albrecht.ecn.purdue.edu/~jauyuen/resume.html">http://albrecht.ecn.purdue.edu/~jauyuen/resume.html</a>, e.g. [10], [9]</p>
Test-Images	<p>A database of about 10000 images containing a wide variety of natural images (people, wildlife, business, art, landscapes, textures, etc.)</p>

### 3.2.12 NETRA (means eye in Sanskrit)

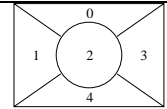
Source	<p>Original demo: <a href="http://maya.ece.ucsb.edu/Netra/">http://maya.ece.ucsb.edu/Netra/</a>  Later version (color image segmentation): <a href="http://maya.ece.ucsb.edu/Netra/index2.html">http://maya.ece.ucsb.edu/Netra/index2.html</a></p> <p>Wei-Ying Ma, B. S. Manjunath, Yaojun Luo, Yining Deng and Xinding Sun at the Department of Electrical and Computer Engineering, University of California at Santa Barbara. NETRA is part of the UCSB Alexandria Digital Library (ADL) project.</p>
Features	<p>Images are automatically segmented into about 6-12 non-overlapping homogeneous regions. The segmentation is based on an edge flow algorithm which uses 'edges' in color and texture features to detect homogeneous regions.</p> <ul style="list-style-type: none"> <li>• <b>Color:</b> Each image region is represented by a subset of colors from a color codebook. Since segmented regions are quite homogeneous, much fewer colors, typically 5-15, are usually sufficient to represent a region.</li> <li>• <b>Texture:</b> The texture features are calculated from Gabor filters, which detects edges and lines, in different scales and directions. A simple representation is constructed using means and standard deviations on the filter responses.</li> <li>• <b>Shape:</b> Boundary pixel coordinates (chain-code), <math>(x_s, y_s)</math>, is extracted for each region and three types of contour features are derived <ul style="list-style-type: none"> <li>– <i>Curvature:</i> The rate of change in tangent direction of the contour, as a function of arc length.</li> <li>– <i>Complex coordinates:</i> <math>Z(s) = (x_s - x_c) + i(y_s - y_c)</math>, where <math>(x_c, y_c)</math> is the region centroid.</li> <li>– <i>Centroid distance:</i> <math>R(s) =  Z(s) </math></li> </ul> </li> </ul> <p>Fourier-based shape descriptors are then calculated from the contour features. The representation can be made rotation invariant (use the amplitude of the Fourier coefficients) and scale invariant (divide amplitude of the coefficients with the DC-component).</p> <ul style="list-style-type: none"> <li>• <b>Spatial location:</b> Two sets of parameters is used: <i>the region centroid</i> <math>(x_c, y_c)</math> and <i>coordinates of its minimum bounding rectangle</i> <math>(x_l, x_r, y_t, y_b)</math> (smallest vertically aligned box which contains the region).</li> </ul> <p>Color, texture, and shape of each region are indexed separately. Vector quantization techniques are used to cluster each of the features (a modified k-mean cluster algorithm is used). The indexing scheme also uses dominant colors in the query image to prune the color space.</p>
Interface	<p>The user starts a search by selecting a category. He then selects an image and makes a query by selecting regions and features, e.g. "retrieve all images that contain regions that have the color of region A, texture of region B, shape of region C, and lie in the upper one-third of the image".</p>
Publications	<p><a href="http://maya.ece.ucsb.edu/Netra/">http://maya.ece.ucsb.edu/Netra/</a> and <a href="http://maya.ece.ucsb.edu/Netra/index2.html">http://maya.ece.ucsb.edu/Netra/index2.html</a>, e.g. [34], [12]</p>
Test-Images	<p>2500 images from Corel photo gallery (<a href="http://www.corel.com">http://www.corel.com</a>)</p>

### 3.2.13 Photobook/FourEyes

Source	<p><a href="http://whitechapel.media.mit.edu/vismod/demos/photobook/index.html">http://whitechapel.media.mit.edu/vismod/demos/photobook/index.html</a></p> <p>Thomas Minka, Alex Pentland, Rosalind W. Picard and Stanley Sclaroff, Media lab, Massachusetts Institute of Technology (MIT)</p>
Features	<p>Aims to calculate information-preserving features, from which all essential aspects of the original image can in theory be reconstructed (allows features relevant to a particular type of search to be computed at search time, giving greater flexibility at the expense of speed). The system is composed of three modules:</p> <ul style="list-style-type: none"> <li>• <b>Appearance:</b> Create a set of basis functions to describe the images by using Karhunen-Loeve transform. One example is <i>eigenfaces</i> to describe face images.</li> <li>• <b>Shape:</b> Based on a variant of the Finite Element Method (FEM). An elastic shape model is aligned to the object using feature points (e.g. edges, corner, or high-curvature points) are the finite element nodes (the model contains a stiffness (<b>K</b>) and a mass (<b>M</b>) matrix). (c.f. <i>snakes</i>). One way to compare objects is to compute the amount of energy needed to align the shape models (if the strain energy required to align two feature sets is relatively small, then the objects are very similar).</li> <li>• <b>Texture:</b> Instead of searching for one "best" model, the approach is to study a variety of models and learn which models perform best. Examples of models are: <ul style="list-style-type: none"> <li>– Reaction-diffusion models (Turing). Models spots and stripes</li> <li>– Markov random fields. Models homogeneous micro-textures</li> <li>– Wold model. Models patterns where periodicity (usually outstanding and is used in the first and quickest step to match query image), directionality and randomness (perceptual complexity) are distinguishing features.</li> <li>– STAR model. Models homogeneous temporal textures.</li> <li>– Flame model. Model flames and fire.</li> </ul> </li> <li>• <b>Text:</b> The images are labeled by <i>power-assisted annotation</i> - the user labels a few images and the system finds similar images using the features above and label those images by itself.</li> </ul>
Interface	<p>The Photobook demos are divided into the modules described above. FourEyes is a more recent version that combine the features. Instead of letting the user choose the relative weight between the features (which by their experience is a non-intuitive and frustrating approach), FourEyes looks at the user's interaction with the data and learns the best features by letting the user give positive and negative example images (relevance feedback).</p>
Publications	<p><a href="http://www-white.media.mit.edu/cgi-bin/tr_pagemaker">http://www-white.media.mit.edu/cgi-bin/tr_pagemaker</a>, e.g. [43], [44]</p>
Test-Images	<ul style="list-style-type: none"> <li>• Appearance: About 3000 images of faces from different angles. (Eyes, cars, 1-D sound signals, and 3-D MRI are also mentioned)</li> <li>• Shape: 60 images of 12 hand-tool objects, 74 tropical fish images. (Rabbits, hands, heads, heart X-rays and some other are also mentioned)</li> <li>• Texture: Over 1000 images from the classical Brodatz texture database, VisTex database (<a href="http://vismod.www.media.mit.edu/vismod/imagery/VisionTexture/">http://vismod.www.media.mit.edu/vismod/imagery/VisionTexture/</a>)</li> </ul>
Comments	<p>Although Photobook itself never became a commercial product, its face recognition technology is now in use by several US police departments.</p>

### 3.2.14 PicSOM

Source	<a href="http://www.cis.hut.fi/picsom/">http://www.cis.hut.fi/picsom/</a> Laboratory of Computer and Information Science, Helsinki University of Technology
Features	<ul style="list-style-type: none"> <li>• <b>Color:</b> Average R-, G-, and B-values are calculated in five separate regions (see figure) of the image resulting in a 15-dimensional color feature vector.</li> <li>• <b>Texture:</b> Also calculated separately for each region. The Y-values of the YIQ color representation are examined and the estimated probabilities for each neighbor pixel being brighter than the center pixel are used as features. This results in a 8-dimensional feature vector for each region, totally 40 parameters.</li> <li>• <b>Edges:</b> Edge detection using <math>3 \times 3</math> Sobel filter on the saturation and intensity components of the HSI color space. The edges is detected in 8 directions and the result is thresholded giving a binary image (the saturation and intensity images are summed together). The image feature is then the histograms on the edge orientation in each region. This gives a 40-dimensional feature vector.</li> <li>• <b>Shape:</b> Three shape features is calculated           <ul style="list-style-type: none"> <li>– Features from the co-occurrence matrix of neighboring edge elements. Eight directions and five regions results in a 320-dimensional feature vector.</li> <li>– Features based on the Fourier transform of the binarized edge image. The 2-dimensional amplitude spectrum is smoothed and down-sampled to form a feature vector of 512 parameters.</li> <li>– The same as the previous feature but the Fourier transform is based on polar coordinates.</li> </ul> </li> </ul> <p>The feature vectors are then separately quantized with a Tree-Structured vector quantization algorithm that uses Self-Organizing Maps (TS-SOMs) at each of its hierarchical level. Thus a tree-structured hierarchical representation of all the images in the database is formed. The complexity of the searches will decrease if the search is made top-down. The relative weights between the features are learned through interaction with the user.</p>
Interface	<p>The approach is based on the relevance feedback technique, in which the human-computer interaction is used to refine subsequent queries to better approximate the need of the user.</p> <p>The user starts by selecting one image and the search engine selects a preliminary set of similar images by comparing their tree-representation. Each unit in the tree has a weight that is equal for all units at the beginning.</p> <p>The user can then refine the search by marking the images with positive and negative values depending on whether the user finds them similar or not. The search engine compares the images in the tree-representation and the features (units) that is most often active gets an increased weight while the weights of the least active units are decreased. The search engine again compares the tree-representations, but now using the new weights. Further refinements can be made until the user is satisfied.</p> <p>Also, the user may at any time switch from the iterative queries to browsing the TS-SOM surfaces simply by clicking on an interesting location of the TS-SOM images.</p>
Publications	<a href="http://www.cis.hut.fi/picsom/publications.html">http://www.cis.hut.fi/picsom/publications.html</a> (or <a href="http://www.cis.hut.fi/~jorma/papers/">http://www.cis.hut.fi/~jorma/papers/</a> or <a href="http://www.cis.hut.fi/~markus/">http://www.cis.hut.fi/~markus/</a> ), e.g. [29], [30], [41]
Test-Images	SUNETS picture archive (4350 images) ( <a href="ftp://ftp.sunet.se/pub/pictures/">ftp://ftp.sunet.se/pub/pictures/</a> ) and Corel Stock photos ( <a href="http://www.corel.com">http://www.corel.com</a> )



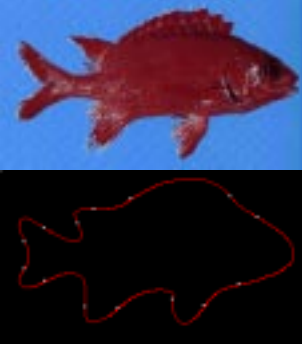
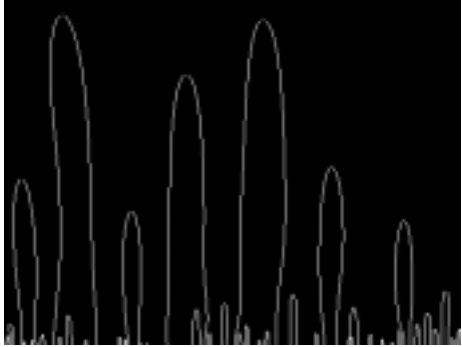
### 3.2.15 QBIC (Query By Image Content)

Source	<p><a href="http://www.qbic.almaden.ibm.com/">http://www.qbic.almaden.ibm.com/</a></p> <p>IBM Research Division, Almaden Research Center</p>
Features	<ul style="list-style-type: none"> <li>• Color: Each axis of the RGB color space is quantized in a predefined number K of levels, giving a color space of <math>K^3</math> cells. After computing the center of each cell in the MTM (Mathematical Transform of Munsell) coordinates, a clustering procedure partitions the space in super-cells. The image histogram represents the normalized count of pixels falling in each super-cell. When performing a query, the query MTM image histogram is matched to the database image histograms. The difference histogram Z is computed, and a similarity measure is given by <math>\ Z\  = Z^T AZ</math>, where A is a symmetrical matrix with A(i,j) measuring the similarity of colors i and j.</li> <li>• Texture: Coarseness, contrast and directionality. Coarseness measure texture scale (average size of regions that have the same intensity), contrast measure vividness of the texture (depends on the variance of the gray-level histogram), and directionality gives the eventual main direction of the image texture (depends on the number and shape of peaks of the distribution of gradient directions). (An improved version of the Tamura texture representation.)</li> <li>• Shape: Area (number of pixels in shape body), circularity (perimeter<sup>2</sup>/Area), eccentricity, major axis orientation (the second order covariance matrix is computed using boundary pixels. Major axis orientation is the direction of the matrix largest eigenvector and eccentricity is the ratio of the smallest and the largest eigenvalue), a set of algebraic moment invariants, and a set of tangent angles around the perimeter.</li> <li>• Text: Keywords</li> </ul>
Interface	<p>QBIC supports queries based on example images, user-constructed sketches and drawings, and selected color and texture patterns. Query on features is allowed on objects (e.g. "find images with red, round objects") or scenes (e.g. "find images that have approximately 30-percent red and 15-percent blue colors") or a combination.</p> <p>The user can select colors and color distributions from a color wheel, select textures from a predetermined selection, and adjust relative weights among the shape features described above.</p> <p>Segmentation into objects can be made either fully automatically (for a restricted class of images) or semiautomatic. In the second approach the segmentation can be made by</p> <ul style="list-style-type: none"> <li>• Flood-fill technique: the user starts by marking a pixel inside the object and adjacent pixels with sufficiently similar color are included (a dynamic threshold is selected by having the user click on both background and object points).</li> <li>• Snakes (active contours): the user draws an approximate contour of the object which then is automatically aligned with nearby image edges.</li> </ul>
Publications	e.g. [17], [15]
Test-Images	Many, see e.g. <a href="http://www-4.ibm.com/software/is/dig-lib/about.html">http://www-4.ibm.com/software/is/dig-lib/about.html</a> . Two demos, a collection of all U.S. stamps before 1995 and a prototype trademark browsing and retrieval site, is available on the QBIC homepage.
Comments	Probably the best-known system and was the first commercial content-based image retrieval system. It is available either in stand-alone form, or as part of other IBM products such as the DB2 Digital Library. It also supports search in video data by using motion, shot detection etc.

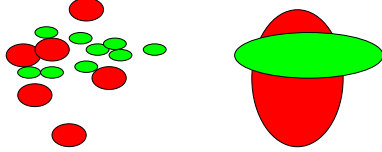
### 3.2.16 SIMPLIcity (Semantics-sensitive Integrated Matching for Picture Libraries)

Source	<a href="http://WWW-DB.Stanford.EDU/IMAGE/">http://WWW-DB.Stanford.EDU/IMAGE/</a> James Ze Wang, Gio Wiederhold, Jia Li and others at Stanford University
Features	<p>The images is classified into a categories which are intended to distinguish semantically meaningful differences, e.g. textured-non textured, indoor-outdoor, objectionable-benign and graph-non graph. The algorithm:</p> <ol style="list-style-type: none"> <li>1. Partition the image into <math>4 \times 4</math> pixel blocks and extracts a feature vector for each block.</li> <li>2. Use the k-mean clustering approach to segment the image into regions.</li> <li>3. The segmentation result is fed into a classifier that decides the semantic type of the image. An image is classified as one of the <math>n</math> pre-defined mutually exclusive and collectively exhaustive semantic classes.</li> <li>4. Features including color, texture, shape and location information are then extracted for each region in the image. The features selected depends on the semantic type of the image.</li> </ol> <p>The image is represented by the collection of features for all of its regions.</p>
Interface	Simply click on an image or specify an ID-number or URL and get similar images.
Publications	<a href="http://WWW-DB.Stanford.EDU/IMAGE/">http://WWW-DB.Stanford.EDU/IMAGE/</a> , e.g. [56]
Test-Images	60,000 images from COREL CD-ROM Collection ( <a href="http://www.corel.com">http://www.corel.com</a> ). The images were segmented and classified into texture and non-texture types (focus on color and texture features).

### 3.2.17 SQUID

Source	<p><a href="http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html">http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html</a></p> <p>Sadegh Abbasi, Farzin Mokhtarian, and Josef Kittler, Centre for Vision, Speech, and Signal Processing, University of Surrey</p>
Features	<p>Curvature features in several scales. They are calculated as follows:</p> <ol style="list-style-type: none"> <li>1. Extract the object boundary contour (chain-code), <math>(x(u), y(u))</math>, by thresholding.</li> <li>2. Calculate a curvature measure defined as the rate of change in tangent direction of the contour, as a function of arc length: <math display="block">\kappa(u) = \frac{\dot{X}(u, \sigma)\dot{Y}(u, \sigma) - \ddot{X}(u, \sigma)\dot{Y}(u, \sigma)}{(\dot{Y}^2(u, \sigma) + \dot{X}^2(u, \sigma))^{3/2}}</math> <p>in several scales <math>\sigma</math>, where</p> <math display="block">X(u, \sigma) = x(u) * g(u, \sigma), \quad Y(u, \sigma) = y(u) * g(u, \sigma)</math> <p>are low-pass version of <math>x(u)</math> and <math>y(u)</math> (<math>g(u, \sigma)</math> is a Gaussian with st.dev. <math>\sigma</math>). <math>\dot{X}(u, \sigma)</math> can be calculated as <math>\dot{X}(u, \sigma) = \dot{x}(u) * g(u, \sigma)</math> etc.</p> </li> <li>3. In each scale, locate curvature zero-crossings (where the curvature is changing between convex and concave). This will give a <i>curvature scale-space</i> (CSS) image as shown below</li> </ol> <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <p>Left: Image and its contour in some scale (zero-crossings marked)  Right: Curvature zero-crossings as function of <math>u</math> (x-axis) and <math>\sigma</math> (y-axis)</p> <ol style="list-style-type: none"> <li>4. Every image is represented by its CSS contour maxima (the local maxima in the figure above), which are above a certain threshold.</li> </ol> <p>This feature description can be made invariant to scale (by normalizing the arc length) and rotation (by using suitable matching). The features works well for easy segmented, non-deformable, 2D-objects.</p>
Interface	There exists at present a demo version with marine animals where the user can choose an image and get similar images. The user can however in theory sketch the contour of an object and use as query image.
Publications	e.g. [38]
Test-Images	over 1000 colored images of marine animals

### 3.2.18 Surfimage

Source	<p><a href="http://www-rocq.inria.fr/imedia/">http://www-rocq.inria.fr/imedia/</a></p> <p>Surfimage is part of the IMEDIA (Image and multimedia indexing, browsing and retrieval) project at INRIA (French National Institute for Research in Computer Science and Control), France</p>
Features	<p>The system extracts both low-level and high-level features:</p> <ul style="list-style-type: none"> <li>• Low-level, global features: Color, orientation, and texture histograms.</li> <li>• Other Low-level: Fourier and Wavelet transforms.</li> <li>• High-level: Eigenimages, flexible images, and image shape spectrum (global histogram of a local curvature measure)</li> </ul> <p>Various similarity metrics are implemented: City-block distance, Minkowski distance, Cosine metric, Hellinger metric, and M-estimators for outline rejection.</p> <p>The relevance feedback approach (see interface) is based on a density estimation in the feature space from the positive and negative examples provided by the user. The features are assumed to be independent and the estimation is performed over each feature component separately. Two models has been used:</p> <ul style="list-style-type: none"> <li>• Gaussian approach: Each feature density is approximated with a Gaussian. A feature with a small standard deviation is more important than a feature with a large standard deviation. Let <math>X</math> be an image and <math>x</math> its feature vector. The images selected are the ones that has the largest probability <math>P(x X \text{ is relevant})</math>.</li> <li>• Parzen estimation: Non-parametric model - each sample contribute with a small, localized density function (e.g. a small Gauss). The images selected are the ones that has the largest value <math>I(x) = -\log(P(x R)) + \log(P(x N))</math>, where <math>R</math> is the set of relevant and <math>N</math> the set of non-relevant images (labeled by the user).</li> </ul> <p>The non-parametric density estimation also solves the problem of multiple queries - the queries can be completely different in terms of content and feature space representation (e.g. faces and cars may be all in the same relevance class, if that is the intention of the user).</p> <div style="text-align: center;">  <p>Left: Parzen window estimation. Right: Gaussian estimation</p> </div>
Interface	<p>Two approaches has been tested. The user can make a query either by selecting an example image, relevant features and metric or use a relevance feedback technique to find the images (the latter is used in the Web-demo).</p>
Publications	<p>[39], [37]</p>
Test-Images	<ul style="list-style-type: none"> <li>• Columbia database of 1440 images consisting of 72 views of 20 objects.</li> <li>• 3670 images from MIT VisTex texture database, BTphoto database of city and country scenes, paintings, and images of people in the lab.</li> </ul>

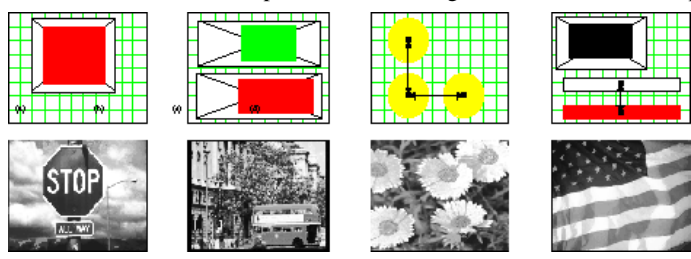
### 3.2.19 Virage Image engine

Source	<p><a href="http://www.virage.com/products/vir-irw.html">http://www.virage.com/products/vir-irw.html</a></p> <p>Virage Technologies, Inc. (formed by Ramesh Jain)</p>
Features	<p>The image is preprocessed with smoothing and contrast enhancement before a set of features is extracted. The features are:</p> <ul style="list-style-type: none"> <li>• <b>Color:</b> Evaluates both dominant color and color variation in the hue, saturation and tint color space.</li> <li>• <b>Composition:</b> Relative locations of colored areas (images with the same colors, the same amount of colors, and colors in the same relative areas are ranked together).</li> <li>• <b>Texture:</b> Granularity, roughness, repetitiveness, etc.</li> <li>• <b>Shape:</b> Evaluates boundary characteristics of distinct shapes to determine a structure value. Because of problems such as lightning effects and occlusion, it relies heavily on shape characterization techniques, rather than local shape segmentation methods.</li> <li>• <b>Text</b></li> <li>• <b>Domain specific:</b> Various useful primitives can be added to the open structure depending on the domain requirements, e.g. eigenfaces for face recognition.</li> </ul> <p>(The system also supports time-dependent features for video queries such as object motion, object discontinuities, scene breaks or cuts, dissolves, fades and wipes.)</p>
Interface	The user can adjust the weights associated with the atomic features according to their own emphasis.
Publications	[2], [19]
Test-Images	Various
Comments	<p>The system is available as an add-on to existing database management systems such as Oracle or Informix (<a href="http://www.informix.com">http://www.informix.com</a>).</p> <p>A high-profile application of Virage technology is AltaVista's AV Photo Finder (<a href="http://image.altavista.com/cgi-bin/avncgi">http://image.altavista.com/cgi-bin/avncgi</a>), allowing Web surfers to search for images by content similarity.</p>

### 3.2.20 VisualSEEK/SaFe (Spatial and Feature query system)

Source	<p><a href="http://www.ctr.columbia.edu/VisualSEEK/">http://www.ctr.columbia.edu/VisualSEEK/</a></p> <p>John R. Smith, Shih-Fu Chang and others at Columbia University, New York. VisualSEEK/SaFe is part of the <b>Advent project</b> (All Digital Video Encoding, Networking, Transmission) which also includes WebSEEK, VideoQ, MetaSEEK, InLumine and WebClip, see <a href="http://www.ctr.columbia.edu/advent/">http://www.ctr.columbia.edu/advent/</a> and <a href="http://www.ctr.columbia.edu/~sfchang/demos.html">http://www.ctr.columbia.edu/~sfchang/demos.html</a></p>
Features	<ul style="list-style-type: none"> <li>• <b>Color:</b> Both global color histograms and regional color descriptions is analyzed. The system extracts color regions in three steps:             <ol style="list-style-type: none"> <li>1. Transform RGB-space into HSV-space.</li> <li>2. Quantize the HSV space into 166 regions.</li> <li>3. The quantized image is then filtered morphologically and analyzed to reveal spatially localized color regions. The region extraction is based on the dominant colors in the region.</li> </ol> <p>The region colors is represented by <i>color sets</i>: a 166 dimensional binary vector which corresponds to a selection of colors. A color set approximates a color histogram by thresholding the histogram. The similarity between two color sets, <math>c_1</math> and <math>c_2</math>, is given by <math>(c_1 - c_2)^T A (c_1 - c_2)</math> where the elements in matrix <math>A</math> denotes the similarity between the colors. The color matching can be made efficient due to the binary property of the color set vector.</p> </li> <li>• <b>Texture:</b> Texture histograms that is derived from spatial-frequency channels (Wavelet transform).</li> <li>• <b>Region features:</b> The region centroids and minimum bounding boxes (smallest vertically aligned rectangle that completely encloses the region) is calculated. Several features based on these parameters is used in the query:             <ul style="list-style-type: none"> <li>– Spatial distance between regions: An optional spatial bound for each region can be designated in which a target region is assigned a spatial distance of zero (this is useful when the user does not care about the exact position of matched regions as long as they fall within a designated area).</li> <li>– Region area (box area) and spatial extent (length of box diagonal)</li> <li>– Region relative location: defined by 2D-strings which is a symbolic projection of the image along the x and y directions.</li> <li>– Special spatial relations: adjacency, nearness, overlap, surround and spatial invariances (translation, rotation).</li> </ul> </li> <li>• <b>Text</b></li> </ul> <p>The intrinsic parameters, such as region color, spatial location and size, is used to reduce the number of candidate images that is evaluated in the final stage where the derived parameters, such as region relative locations and special spatial relations, is analyzed. This is because the latter evaluations have the highest complexity.</p>

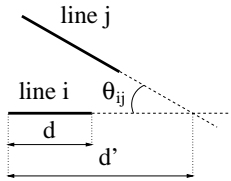
*continued on next page*

<i>continued from previous page</i>	
Interface	<p>The user sketches the regions, positions them on the query grid and assigns them properties of color, size and absolute location. The user may also assign boundaries for location and size of the regions. Both absolute and relative placement of the regions can be used in the query.</p>  <p>Example queries: 1) Region with absolute location, 2) Two regions with absolute locations, 3) Multiple regions with relative locations, 4) Multiple regions with both absolute and relative locations.</p> <p>VisualSEEk has two Web-demos:</p> <ul style="list-style-type: none"> <li>• <b>SaFe</b> allows you to search for images by spatial content using the SaFe Java applet.</li> <li>• <b>CBVQ</b> (content-based visual query system) allows you to search for images by color histograms (automatically or manually defined), texture and color composition. You may start a search using any image from the Web by providing its address.</li> </ul>
Publications	<a href="http://www.ctr.columbia.edu/advent/home-full.html">http://www.ctr.columbia.edu/advent/home-full.html</a> and <a href="http://www.ctr.columbia.edu/~jrsmith/publications.html">http://www.ctr.columbia.edu/~jrsmith/publications.html</a> , e.g. [54], [52], [53]
Test-Images	A test-bed of 12000 color images and a collection of over one million images and videos from the World-Wide Web.

### 3.2.21 WebSEEK

Source	<p><a href="http://www.ctr.columbia.edu/webseek/">http://www.ctr.columbia.edu/webseek/</a></p> <p>John R. Smith, Shih-Fu Chang and others at Columbia University, New York, see VisualSEEK/SaFe for more information.</p>
Features	<p>The system searches on the Web for images and videos (by looking for the file extensions .jpg, .mov, .gif, .mpeg, etc.). The system then extracts keywords from the URL and hyperlinks, e.g. the URL 'http://www.arch.columbia.edu/images/animals/dog.jpg' gives the keywords 'images', 'animals' and 'dog'. Next the system creates a histogram on the keywords to find the most frequent ones. The words with multiple meanings make poor keywords and are removed from the set, the rest is used to classify the images. The words are manually put into classes and subclasses, which makes up a tree of keywords (an image can be found on several leaves if it has several useful keywords).</p> <p>Global color histograms are also computed from each image which can be used in a query (see VisualSEEK for more information).</p>
Interface	The user can search for images either by walking in the keyword tree or using the keywords directly. The system then displays a selection of images within the category, which users can then search by color histogram similarity. The histogram can be modified either manually or by relevance feedback facilities.
Publications	[51]
Test-Images	More than 650,000 images and videos from the web.

### 3.2.22 Huet and Hancock

Source	Benoit Huet, Edwin R. Hancock and others at University of York, England.
Features	<p>Matching objects or images is based on <b>attributed relational graphs</b>. Each shape in the database is a pattern of line-segments (Each node in the graph contains a line-segment).</p> <p>Two attributes (features) is calculated between pairs of line-segments: <b>Relative angle</b> <math>\theta_{ij}</math> and <b>relative position</b> <math>\vartheta_{ij} = \frac{1}{\frac{1}{2} + \frac{d}{a}}</math> (see figure). These gives a two-dimensional feature vector <math>x_{ij} = (\theta_{ij}, \vartheta_{ij})</math>. The line-segment <math>i</math> is represented by a set of feature vectors <math>\{x_{ij}\}_{j=1}^J</math> and by a (2D) histogram <math>h_i(\theta, \vartheta)</math> calculated from the feature set.</p>  <p>The feature sets are used to match two line-segments from two different images. The probability that a line-segment <math>i</math> in image 1 is similar to line-segment <math>m</math> in image 2 is either based on finding matching pairs of feature vectors from the sets <math>\{x_{ij}\}_{j=1}^J</math> and <math>\{x_{mn}\}_{n=1}^N</math> or matching the histograms <math>h_i</math> and <math>h_m</math>.</p> <p>The match between two images is then based on finding good matches between a number of line-segment pairs.</p> <p>The features <math>\theta</math> and <math>\vartheta</math> is invariant to rotation, translation and scale which makes the total match invariant to the same properties.</p> <p>The algorithm is tested against robustness to extra lines, missing lines, split lines, segment end-point errors and combined errors. It is most sensitive to missing lines.</p>
Interface	-
Publications	<a href="http://www-users.cs.york.ac.uk/~huetb/">http://www-users.cs.york.ac.uk/~huetb/</a> , e.g. [21], [23], [22].
Test-Images	Database containing 2500 line patterns (each containing over a hundred lines) extracted from real-world imagery - aerial images, trademarks and logos.

### 3.2.23 Jain and Vailaya

Source	Anil K. Jain, Aditya Vailaya and others at Michigan State University
Features	<p>The search is divided into a fast pruning stage and a more complex object matching stage:</p> <ol style="list-style-type: none"> <li>1. <b>Fast pruning stage:</b> The database is reduced using a set of scale-, rotation-, and translation invariant features: <ul style="list-style-type: none"> <li>• <b>Edge angles:</b> Global histogram of the edge directions. The edges is extracted using Canny edge detector. The match is made invariant to rotation by shifting the histogram bins (the histogram is smoothed to get rid of the quantization effect).</li> <li>• <b>Shape:</b> The global image shape is described using seven invariant moments.</li> </ul> </li> <li>2. <b>Object match:</b> Deformable templates is used (the template is attracted and aligned to the salient edges in the image via a directional edge potential field). The query image is used as the prototype, which is deformed to match the test trademarks. The initial pose parameters of the template (position, scale, and orientation) are estimated using the generalized Hough transform. An energy function is then calculated to evaluate the quality of the match.</li> </ol>
Interface	-
Publications	<a href="http://www.cse.msu.edu/~vailayaa">http://www.cse.msu.edu/~vailayaa</a> , e.g. [26], [55]
Test-Images	Trademark image database containing 1100 binary images. The evaluation was made by letting 5 persons to retrieve images from the same database and the same queries (it took each subject between 1 and 2 hours to find the top ten retrievals for five query images).

## 4 Review articles

- [1] write about applications and methods for image and video archives and mentions QBIC, Virage, Chabot, CANDID, Photobook and VisualSEEk (and some video retrieval systems)
- [4] describes the leading techniques until 1996. Contains a short description of QBIC, Virage, Excalibur, ALISA, Illustra, Chabot and CIIR.
- [13] write about applications, needs, current techniques, and standards (e.g. MPEG-7) for image and video retrieval. Reviews QBIC, Virage, Excalibur, Photobook, Chabot, VisualSEEk, MARS, Informedia, Surfimage, and Synapse. Quite thorough.
- [35] is a nice survey on indexing methods for image retrieval and reviews QBIC, Photobook, CANDID, Chabot, Piction, ART-MUSEUM and some others. It contains an table overview similar to the one in this paper.
- [20] reviews QBIC, Advent and Virage. Thinks Advent is the best one of the three.
- [45] is a survey of the technical achievements in the research area of Image Retrieval. Reviews QBIC, Virage, RetrievalWare, Photobook, VisualSEEk/WebSEEk, Netra, MARS, and mentions some others.
- [57] is a survey on content-based multimedia retrieval (image, video, audio)

## 5 Acknowledgment

This work was supported by the Swedish Foundation for Strategic Research, project VISIT - Visual Information Technology. I would also like to thank my supervisor Gösta Grandlund for useful insights.

## References

- [1] Kjersti Aas and Line Eikvil. A survey on: Content-based access to image and video databases. Report 915, Norwegian Computing Center, March 1997. URL: <http://www.nr.no/home/kjersti/video.html>.
- [2] Jeffrey R. Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh Jain, and Chiao fe Shu. The virage image search engine: An open framework for image management. In *Proceedings SPIE Storage and Retrieval for Image and Video Databases*, volume 2670, pages 76–87, 1996.
- [3] Mark F. Bear, Barry W. Connors, and Michael A. Paradiso. *Neuroscience: Exploring the Brain*. Williams & Wilkins, 1996. ISBN 0-683-00488-3.
- [4] Glenn Becker. *Information in Images, chap 6: Content-based Query of Image Databases*. Thomson Technology Labs, 1997. Online book, URL: <http://www.thomtech.com:80/mmedia/tmr97/tmr97.htm>.
- [5] Serge Belongie, Chad Carson, Hayit Greenspan, and Jitendra Malik. Color- and texture-based image segmentation using em and its application to content-based image retrieval. In *Proceedings of the Sixth International Conference on Computer Vision*, January 1998.

- [6] Peter Bock. Alisa: Adaptive learning image and signal analysis. In *Proceedings of the SPIE Applied Imagery Pattern Recognition Conference*, Washington DC, October 1998. URL: <http://www.seas.gwu.edu/~pbock/>.
- [7] Brian Carlson. Web still & video image content search & retrieval. *Advanced imaging*, pages 34–36, 60, April 1997.
- [8] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. 1999. Submitted to PAMI.
- [9] Jau-Yuen Chen, Charles A. Bouman, and Jan P. Allebach. Fast image database search using tree-structured vq. In *Proceedings of the International Conference on Image Processing*, Santa Barbara, CA, October 26-29 1997. URL: [http://dynamo.ecn.purdue.edu/~bouman/publications/pub\\_database.html](http://dynamo.ecn.purdue.edu/~bouman/publications/pub_database.html).
- [10] Jau-Yuen Chen, Charles A. Bouman, and Jan P. Allebach. Multiscale branch and bound image database search. In *Proceedings of IS&T/SPIE Conf. on Storage and Retrieval for Image and Video Databases V*, volume 3022, pages 133–144, San Jose, CA, February 13-14 1997. URL: [http://dynamo.ecn.purdue.edu/~bouman/publications/pub\\_database.html](http://dynamo.ecn.purdue.edu/~bouman/publications/pub_database.html).
- [11] Marc Davis. Media streams: An iconic visual language for video representation. *Readings in Human-Computer Interaction: Toward the Year 2000*, ed. Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton, and Saul Greenberg, 2nd ed., San Francisco: Morgan Kaufmann Publishers, Inc., pages 854–866, 1995. URL: <http://www.menlo.com/folks/davis/>.
- [12] Yining Deng and B.S. Manjunath. An efficient low-dimensional color indexing scheme for region-based image retrieval. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 6, pages 3017–3020, 1999. URL: <http://maya.ece.ucsb.edu/~deng/>.
- [13] John P. Eakins and Margaret E. Graham. Content-based image retrieval. a report to the jisc technology applications programme. Technical report, Institute for Image Data Research, University of Northumbria at Newcastle, January 1999. URL: <http://www.unn.ac.uk/iidr/research/cbir/report.html>.
- [14] Excalibur visual retrievalware: Technical summary. URL: <http://www.excalib.com/products/vrw/vrwtechsum.html>.
- [15] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Perkovic, and R. Barber. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.
- [16] Judy Feder. Towards imaging content-based retrieval for the world-wide web. *Advanced imaging*, 11(1):26–28, 1996.
- [17] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video content: The qbic system. *Computer*, pages 23–32, September 1995.
- [18] G. H. Granlund. The complexity of vision. *Signal Processing*, 74(1):101–126, April 1999. Invited paper.
- [19] Amarnath Gupta. Visual information retrieval technology: A virage perspective. Revision 4, Virage Image Engine API Specification, Virage Inc., February 11 1997. URL: <http://www.virage.com>.

- [20] Qin He. Comparison of content-based image retrieval systems. Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, Spring 1996. URL: <http://www.lis.uiuc.edu/~q-he1/lis429/project.html>.
- [21] Benoit Huet, Andrew D. J. Cross, and Edwin R. Hancock. Shape retrieval by inexact graph matching. In *IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, volume 1, pages 772–776, Florence, Italy, June 7-11 1999. URL: <http://www-users.cs.york.ac.uk/~huetb/>.
- [22] Benoit Huet and Edwin R. Hancock. Inexact graph retrieval. In *IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL-99)*, pages 40–44, Fort Collins, Colorado, June 22 1999.
- [23] Benoit Huet and Edwin R. Hancock. Sensitivity analysis for object recognition from large structural libraries. In *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV99)*, volume 2, pages 1137–1143, Kerkyra, Greece, September 20-27 1999.
- [24] D. P. Huijismans, Michael S. Lew, and Dee Denteneer. Quality measures for interactive image retrieval with a performance evaluation of two 3x3 texel-based methods. In *ICIAP97*, 1997. URL: <http://www.wi.leidenuniv.nl/~huijsman/>.
- [25] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In *Proceedings of SIGGRAPH 95*, Los Angeles, August 6-11 1995. ACM SIGGRAPH, New York. URL: <http://www.cs.washington.edu/homes/salesin/abstracts.html>.
- [26] Anil K. Jain and Aditya Vailaya. Shape-based retrieval: A case study with trademark image databases. *Pattern Recognition*, 21(9):1369–1390, 1998. URL: <http://www.cse.msu.edu/~vailayaa/>.
- [27] Patrick Kelly and Michael Cannon. Experience with candid: Comparison algorithm for navigating digital image databases. In *Proceedings of the 23rd AIPR Workshop on Image and Information Systems: Applications and Opportunities*, volume 2368, pages 64–75, Washington, D.C., October 12-14 1994. SPIE. URL: <http://www.c3.lanl.gov/~kelly/CANDID/pubs.shtml>.
- [28] Patrick M. Kelly, Michael Cannon, and Donald R. Hush. Query by image example: the candid approach. In *Storage and Retrieval for Image and Video Databases III*, volume 2420, pages 238–248. SPIE, 1995. URL: <http://www.c3.lanl.gov/~kelly/CANDID/pubs.shtml>.
- [29] J. T. Laaksonen, J. M. Koskela, and E. Oja. Picsom - a framework for content-based image database retrieval using self-organizing maps. In *Proceedings of SCIA'99, to be published*, Kangerlussuaq, Greenland, June 1999. URL: <http://www.cis.hut.fi/picsom/publications.html>.
- [30] Jorma Laaksonen, Markus Koskela, and Erkki Oja. Content-based image retrieval using self-organizing maps. In *Proceedings of VISual'99, to be published*, Amsterdam, The Netherlands, June 1999. URL: <http://www.cis.hut.fi/picsom/publications.html>.
- [31] Michael S. Lew, D. P. Huijismans, and Dee Denteneer. Content based image retrieval: Optimal keys, texture, projections, or templates. In *Proceedings of the 1th International Workshop on Image Databases and Multi-Media Search*, pages 27–34, 1996. URL: <http://www.wi.leidenuniv.nl/home/lim/lim.html>.

- [32] Michael S. Lew and Nies Huijsmans. Information theory and face detection. In *Proceedings of the International Conference on Pattern Recognition*, pages 601–605, August 25-30 1996. URL: <http://www.wi.leidenuniv.nl/~mlew/>.
- [33] Michael S. Lew, Kim Lempinen, and Nies Huijsmans. Webcrawling using sketches. In *Proceedings VISUAL97 conference*, pages 77–84, December 15-17 1997. URL: <http://www.wi.leidenuniv.nl/~mlew/>.
- [34] W. Y. Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. In *Proceedings of the IEEE International Conference on Image Processing*, 1997. URL: <http://maya.ece.ucsb.edu/Netra/>.
- [35] M De Marsicoi, L Cinque, and S Levialdi. Indexing pictorial documents by their content: a survey of current techniques. *Image and Vision Computing*, 15:119–141, 1997.
- [36] Sharad Mehrotra, Yong Rui, Michael Ortega-Binderberger, and Thomas S. Huang. Supporting content-based queries over images in mars. In *Proceedings of IEEE Int. Conf. Multimedia Computing and Systems*, pages 632–633, June 3-6 1997. URL: <http://research.microsoft.com/users/yongrui/html/publication.html>.
- [37] Christophe Meilhac and Chahab Nastar. Relevance feedback and category search in image databases. In *IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, Florence, Italy, June 7-11 1999. URL: <http://www-rocq.inria.fr/imedia>.
- [38] Farzin Mokhtarian, Sadegh Abbasi, and Josef Kittler. Robust and efficient shape indexing through curvature scale space. In *Proceedings of the British Machine Vision Conference*, pages 53–62, Edinburgh, UK, 1996. The Institution of Electrical Engineers. URL: <http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html>.
- [39] Chahab Nastar, Matthias Mitschke, Christophe Meilhac, and Nozha Boujema. Surfimage: a flexible content-based image retrieval system. In *ACM-Multimedia*, Bristol, England, September 12-16 1998. URL: <http://www-rocq.inria.fr/imedia>.
- [40] Virginia E. Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9), September 1995. URL: <http://http.cs.berkeley.edu/~ginger/>.
- [41] Erkki Oja, Jorma Laaksonen, Markus Koskela, and Sami Brandt. Self-organizing maps for content-based image database retrieval. In *Proceedings of WSOM'99*, Espoo, Finland, July 1999. URL: <http://www.cis.hut.fi/picsom/publications.html>.
- [42] Michael Ortega, Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, and Thomas S. Huang. Supporting similarity queries in mars. In *Proceedings of ACM Multimedia*, pages 403–413, November 8-14 1997. URL: <http://research.microsoft.com/users/yongrui/html/publication.html>.
- [43] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. In *SPIE Storage and Retrieval Image and Video Databases II*, number 2185, San Jose, February 6-10 1994. URL: <ftp://whitechapel.media.mit.edu/pub/tech-reports/TR-255-ABSTRACT.html>.

- [44] Rosalind W. Picard. A society of models for video and image libraries. *IBM Systems Journal*, 35(3,4):292–312, 1996. URL: [http://www-white.media.mit.edu/cgi-bin/tr\\_pagemaker](http://www-white.media.mit.edu/cgi-bin/tr_pagemaker) or <http://www.research.ibm.com/journal/sj/mit/sectiona/picard.html>.
- [45] 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*, 10:39–62, March 1999. URL: <http://research.microsoft.com/users/yongrui/html/publication.html>.
- [46] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Content-based image retrieval with relevance feedback in mars. In *Proceedings of IEEE International Conference on Image Processing*, pages 815–818, October 26-29 1997. URL: <http://research.microsoft.com/users/yongrui/html/publication.html>.
- [47] Yong Rui, Thomas S. Huang, Sharad Mehrotra, and Michael Ortega. Automatic matching tool selection via relevance feedback in mars. In *Proceedings of The 2nd International Conference on Visual Information Systems*, pages 109–116, December 15-17 1997. URL: <http://research.microsoft.com/users/yongrui/html/publication.html>.
- [48] Yong Rui, Thomas S. Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Tran on Circuits and Systems for Video Technology , Special Issue on Segmentation, Description, and Retrieval of Video Content*, 8(5):644–655, September 1998. URL: <http://research.microsoft.com/users/yongrui/html/publication.html>.
- [49] Chi-Ren Shyu, Carla Brodley, Avi Kak, Akio Kosaka, Alex M. Aisen, and Lynn S. Broderick. Local versus global features for content-based image retrieval. In *Proceedings IEEE Workshop of Content-Based Access of Image and Video Libraries*, pages 30–34, June 1998. URL: <http://RVL.www.ecn.purdue.edu/RVL/CBIR/CBIRPublications.html>.
- [50] Chi-Ren Shyu, Carla Brodley, Avi Kak, Akio Kosaka, Alex M. Aisen, and Lynn S. Broderick. Assert, a physician-in-the-loop content-based image retrieval system for hrct image databases. *Computer Vision and Image Understanding*, July 1999, to appear. URL: <http://RVL.www.ecn.purdue.edu/RVL/CBIR/CBIRPublications.html>.
- [51] John R. Smith and Shih-Fu Chang. Searching for images and videos on the world-wide web. Report 459-96-25, Center for Telecommunication Research, Department of Electrical Engineering and Center for Image Technology for New Media, Columbia University, New York, N.Y. 10027, August 19 1996. URL: <http://www.ctr.columbia.edu/WebSEEk/about.html>.
- [52] John R. Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. *ACM Multimedia '96*, November 1996. URL: <http://www.ctr.columbia.edu/~jrsmith/publications.html>.
- [53] John R. Smith and Shih-Fu Chang. Querying by color regions using the visualseek content-based visual query system. In *M. T. Maybury, editor, Intelligent Multimedia Information Retrieval, International Joint Conference on Artificial Intelligence, 1997*. URL: <http://www.ctr.columbia.edu/~jrsmith/publications.html>.

- [54] John R. Smith and Shih-Fu Chang. Integrated spatial and feature image query. *Multimedia System Journal*, 7(2):129–140, March 1999. URL: <http://www.ctr.columbia.edu/~jrsmith/publications.html>.
- [55] Aditya Vailaya, Yu Zhong, and Anil K. Jain. A hierarchical system for efficient image retrieval. In *13th International Conference on Pattern Recognition*, Vienna, August 25-30 1996. URL: <http://www.cse.msu.edu/~vailayaa/>.
- [56] James Ze Wang, Jia Li, Desmond Chan, and Gio Wiederhold. Semantics-sensitive retrieval for digital picture libraries. *D-Lib Magazine*, 5(11), November 1999. ISSN 1082-9873, URL: <http://www.dlib.org>.
- [57] Atsuo Yoshitaka and Tadao Ichikawa. A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81–93, January/February 1999.