Crossing the divide between computer vision and data bases in search of image databases

Arnold W.M. Smeulders, Martin L. Kersten, Theo Gevers
ISIS, Department of Computer Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, the Netherlands
{smeulders, mk, gevers}@wins.uva.nl

Abstract
Image databases call upon the combined effort of computing vision and database technology to advance beyond exemplary systems. In this paper we charter several areas for mutually beneficial research activities and provide an architectural design to accommodate it.

keywords
databases, computer vision, invariant features, image databases

1 INTRODUCTION

Image databases, what once was a quiet research topic, has moved to the center of attention. This is due to the fact that practically all imaging devices now deliver their images digitally; the standard storage capacity has surpassed the threshold of storing more than a thousand images; and Internet has introduced a strongly visual component in the domain of the computer. No wonder that visual search, query by visual example and visual browsing are urgent scientific questions now.

One means to effectively search for a specific image is to use a catalogue (http://ipix.yahoo.com) or use textual information and a standard web browser to located a prospective site for further browsing. Such caption and text based search for pictorial information should always be explored, as captions may be very informative on the picture content but can not capture anything of the image content outside the intention of the writer. In this paper we make one step beyond text-based search for picture information: we search for image data bases.

The prime research communities involved in advancing the state of the art in visual information systems and image data bases are the database and computing vision communities. The database community brings its wealth of experience in organizing large amounts of data with an easy to use retrieval language (SQL) provided that the data are structured. The computing vision community brings its wealth of experience in a partial understanding what makes an image come about and computational techniques to recognize objects in real world scenes in not yet well delineated domains.

As a consequence, attempts from both sides have produced prototypes of limited functionality, such as Photobook [16], VisualSeek [22], Virage [15] and QBIC [19]. They either provide access to image archives based on contextual information and-or a measure of similarity based on color, shape, and texture features. These search mechanisms may sometimes lead to some success, depending on the ability of the features to capture the essence of the visual domain at hand. Rarely one sees an integration of picture content search mechanisms in a database context.

In the image retrieval arena, the efficiency of the integration with database techniques is mostly missing, resorting to file-based search. Also the trustworthiness of the query answering scheme is underdeveloped. The methods published so far have not been shown to scale to more than a few Hundred images or beyond a closed application (query) domain. Tight integration with a full-blown database systems is not addressed. e.g. QBIC is loosely linked with DB2.

In this paper we charter a route towards combining the expertise from both areas to make further progress. The approach taken complements those reported in [15] and [14] by not protruding either field in isolation, but to identify their roles and synergetic means. In particular, we challenge the computing vision community to develop the database scheme to model *distinctive image features* and effective algorithms to compute them for large collections in real time. From the database community we expect an *incremental indexing scheme* based on differentiation rather than sorting by commonalties. This provides the skeleton framework for handling the intrinsic continuous nature of sensory information, which can be subsequently searched using proximity queries.

The remainder of this paper is organized as follows. In section 2 we review the contributions and challenges of computer vision to deal with image databases. Likewise, in section 3 we touch upon progress made in the database arena. An overview of our experimental architectural design is given in section 4. Research questions to bridge the chasm are summarized in section 5.

2 COMPUTER VISION SUPPORT FOR IMAGE RETRIEVAL

It is clear that the state of the art in computer vision does not permit the automatic interpretation of an arbitrary scene. There is no general computational method which enables the analysis of an arbitrary scene, and it is very unlikely such an algorithm will be found (shortly). The problems are twofold.

A picture of one object may assume a million different data fields depending on the variety in illumination circumstances, the shadows other objects cast on the object, the magnification, orientation and rotation with which the object falls on the field of view, and the projective deformation of the object. Each time the field of view of the object or the illumination patterns is different. And, one object is each time represented by a different data field. *Thesis 1: As a consequence, the first and foremost problem of computer vision is that there are a million different image data arrays which depict the same object.* The features computed from images are based on mathematical equivalence of illumination and reflectance patterns rather than on semantic equivalence of images. A user of an image database containing an object and a tumbled version of the same object, would expect that both images were retrieved as the semantic is exactly the same. From a computer vision point of view, the images differ in all intensity data values. Where the image data themselves are so variable, depending on accidental aspect as viewer position, and surrounding objects, the search is for other ways to capture the contents of the image irrespective of the pose of the object.

The second problem is that an interpretation of image has no unique meaning. The meaning and interpretation of an image cannot be derived from the data alone. Contextual information and knowledge of the world is essential to deliver an interpretation of the picture. For example, an X-ray of the thorax may have "a medical image" as its prime interpretation for the general public. For a family doctor, the same X-ray may have an interpretation "something serious the matter", whereas the radiologist would never use these two interpretations of the image. For the radiologist the interpretation of medical image is so obvious that such a denomination would never occur. Similarly, as soon as the eyes fall on the X-ray, the interpretation fixes at a list of diseases and possible interpretations. The same interplay between picture and interpretation exist for general pictures, say an outdoor scene. There the context may induce the selection of certain objects to name first in the interpretation of the scene. But that selection is viewer dependent. Also, from a strictly data processing point of view no visual evidence may be present in the image to supporting the best description of the picture. As an example, no sun may be visible to describe a scene by "a sunny day". *Thesis 2: As a consequence, an interpretation of a picture heavily depends on context, purpose and domain; an interpretation is rarely unique and visual evidence may be absent in the actual denomination; hence the definition of the pictorial search outcome is poorly defined.*

Theses 1 and 2 express the impossibility of finding a generic solution to all vision problems by one algorithm. Still the computer vision community has made progress in recent years in accessing the visual content of limited domains under limited circumstances in a wealth of useful and necessary techniques.

2.1 Key notions of computer vision for image retrieval

In this section, we address some selected notions of computer vision in view of their potential role in image databases. We discuss image processing functions used in the preprocessing of image data, as well as the notion of segmentation by which the extent of an object is identified in the image data field. We discuss the role of weak segmentation in image retrieval as well as the image-based features. We demonstrate the limited use of models in general image search engines, and finally come to discuss the role of invariance.

Image processing functions map image data fields into another image data field. Common types of these preprocessing functions undo the deformation of the image induced by the sensor (restoration); to enhance the presence of geometrical elements in the image (enhancement); to compute the numerical differential of the image to indicate where boundaries in the image are (gradient field); to estimate velocity in a stream of image data (optic flow); or to condense their representation using a lossy function (compression).

Segmentation is a computational method to assess the set of points in an image which represent one object in the scene. An object may be a large variety of things: a toy, a tree in a forest, a pond, a tumor region, ink in the water but also a movement pattern in video such as waving goodbye and parking a car. An object in this context is represented by a contiguous part of the image data field, denoted as *object*. It may also be represented by one or more *patches* in the data field. Many different computational techniques for segmentation exist, none of which is capable of handling any reasonable set of real world images. Segmentation is complicated because objects may be partially occluded from sight by the presence of other objects, or hard to distinguish in a surrounding of other objects. As discussed in the previous section, computational techniques for segmentation are seriously complicated by scene depending conditions.

To address the issue of partial object occlusion, but also to make a start in the search for effective image indices, *weak segmentation* of the image may be sufficient to identify the presence of an object in an image. Where a house may be identified by its rooftop, weak segmentation delivers a set of patches part of the object's image (but not the complete image of the object). Weak segmentation is useful for image retrieval when the assumption holds that some patches in the picture are decisive to identify the object even when parts of the object are invisible due to occlusion behind another object. To give one more example, soccer players of one team can all be identified in the image by patches with their specific color combinations. To enhance the difference with the fans in the audience the patch may be extended to include some of the green for the grass they play on. Patches of the object's image are enough to identify it. We will exploit the weak segmentation property later on.

A *feature* captures some aspect of an image, a point in the image, a patch in the image, or -after segmentation- an object in the image.

A *histogram of feature values* of an image is the frequency range of feature values in the image. For the purpose of retrieval by image content we are rarely interested in the statistics of all intensity or color values of all data points in the image as the image may be an accidental view of more than one object. We rather imply the storage of aspect values of some patches in the object's image. These aspect values are computed as features of salient patches to be addressed by value. Hence they are stored in sparsely occupied histograms. Note that the histogram of an incomplete segmentation might contain the feature values of the (salient) patches of more than one object. The histogram may contain feature values of other portions in the image, as well as some portions of the histogram may be missing when part of the central object is occluded from sight.

In *geometrical model-based* image analysis a deformable model is matched to the image data field. The state vector of the model captures the pose, orientation and the goodness of fit of the object. Preparing a model and developing a robust model match procedure requires considerable development effort. Results of model-based image analysis are usually suited for narrow domain of image images. That is, they work for one specific application, one specific sensor and one specific set of questions. The conclusion for image retrieval is on the use of geometrical models is that they are useful only for *a narrow picture domain*. Examples of such narrow domains are an encyclopedia of all brands of roses, or the archive of X-ray photographs of the thorax.

As is the case with geometric models, the development of *symbolic reasoning* models requires a considerable amount of work, suited for a small image domain, impossible to repeat for a broad range of objects. It incorporates the common knowledge we acquire in the first 10 years of our life. Constraint resolution as the way to incorporate domain knowledge is rarely suitable for image databases due to the required complexity of the model. The conclusion for image databases is to use a logical model for a narrow domain of images such as the interpretation of maps. The logical model for the general class of all images would be so big and complex that it would become unpractical to handle if such a default knowledge model of the world can be defined at all.

A key issue in image retrieval is *invariance* of features. A feature of an object is invariant if and only if the value of the feature remains the same after an alteration in the conditioning of the scene. As discussed in the previous section, variations in illumination and the influence of the light source in the data scene requires the consideration of some form of invariant features, insensitive to the undesired changes in the scene. As an example, when the object is at an unknown distance to the sensor, scale, translation and rotation invariant features may be in order to enable search for the same object. For outdoor scenes, the most difficult invariance to handle computationally originates from the great number of viewpoints one can take of an object. This requires viewpoint invariant features in order to recognize the object in the image (and not the image itself) as identical.

Paradoxically, with invariance a *warning* for their use should be posted. In the design of features, invariance should be as light as possible as all unnecessary invariance reduces the discriminatory power of the feature. Also, in its use, if illumination invariance is not necessary -when all pictures are standardized recordings of paintings- that invariance should not be included in the feature query set as it reduces the selective power.

2.2 Anatomy of a visual search engine

The make the state of the art in computer vision-based query engines more concrete abstract consider the PictoSeek system [18] as a typical example for pictorial search engines. The various systems as cited above differ in the user interface, the type and implication of their internal feature set and in the way the result is presented but not in their general system architecture.

The system consists of the following computational blocks:

- To define the object of query, an image is recorded or selected from a repository. The aim to find a similar image in the database. Note that "similar image" may imply a partially identical image (as in the case of finding stamps), or a partially identical object in the image (as in the case of a stolen goods database), or a similar styled image (as in the case of a fashion design support system).
  Some of the existing systems characterize the query image by parts with a typical average color. In the QBIC system [19], the user is free to sketch a region in the image with the preferred color. In the Picasso system [17], a sketch of the query object is given and the spatial arrangement of object is taken into account. The PictoSeek and Virage systems let the user select an example image to search for.
- The essence of the query image is captured in a set of features. These features may cover each aspect of the image data, a measurement of intensity, shape, color or texture, movement or model adherence. A key issue is that the collection of features to use for querying is selective and an essential part of the query formulation. Feature extraction may be preceded by image preprocessing steps, model-matching or symbolic analysis of the image as well as a segmentation or a weak segmentation step. In all cases of image retrieval the process results in a condensation of the visual information in feature sets.
  Usually the features are calculated after a weak segmentation. That is the feature values are computed from a few (salient) patches in the image. The size of the patch is reduced to one point or a pair of two close points.
  The Photobook system [16] and QBIC systems as typical examples con-

centrate on RGB-color features of selected regions in the image data field. The PictoSeek system concentrates on color features measured from salient patches in the image. The salient patches are the result of a weak segmentation procedure on the color shape pattern in the image. The size of the patch is reduced to one point or a pair of two close points. The color features are salient points and point pairs enables the identification of colored objects from just a few data points and their color values in the image.

- The images in the set to be queried have been indexed by the same features during insertion of the image into the system, computed by the same processing steps to arrive at an identical feature description for each image. Processing may be different to normalize for a different sensor and different recording circumstances. Where possible, the restoration step will be specific for each different setting of the recordings. Moreover, as the algorithm reflects a computational method, the rule should be that the feature set derived from the image is independent from the implementation but refers to the essential sensory aspect of the object. The feature set is stored as an index for similarity comparison at run time.
- As discussed before, attention is to be paid to the desired classes of invariance. For each image retrieval query a proper definition of the desired invariance is essential. A concise list of the most important invariance properties is:
  Is the search for objects in different orientations and scales?
  Is the search for objects in a large variety of scenes?
  Is the search for objects in other kind of light?
  Is the search for objects from different viewpoints?
  Is the search for an object irrespective occlusion?
  Note that these invariances can each be turned off or on. As an example, in the search on a database for stamps, the viewpoint invariance will best be switched off as the recording of stamps is usually in frontal view only. This holds also for art. For real world data the viewpoint invariance is a desirable property of the query as it does not ask for the object to be in precisely the same view.
  In the current state of the art of query engines, the explicit mentioning of invariance receives little attention. Invariance is usually handled by making a system specific for one application such as stamps or art. In large databases, the availability of a choice of invariances at the time of the query definition is essential.
  In the PictoSeek system both viewpoint invariant color and shape features, as well as illumination invariant features are included. The desired type of invariance will determine the brand of features used in the query.
- The features are computed for each of the salient patches in the image and captured in sparsely occupied histograms. These histograms indicate the presence of color and shape characteristic for the object.
  To permit faster access, in the PictoSeek system the histograms of each

image is accessed via a hash table. The color histogram of each salient point pair (thus containing a color along each of the two axes) is summarized in 6 x 6 bins = 36 values. If a bin contains the color, the hash table sets the corresponding bit. The hash table thus is 36 bits wide, permitting a useful compromise between distinction of colors and speed of access. Other visual engines will contain similar hash tables, but details are not always publicly available.

- The actual query consists of a similarity search for the element in the queried set closest to the query image. As both the query images as the data set is captured in feature values, the similarity function operates between the feature sets. Again, the make the query useful, attention has to be paid to the selection of the similarity function. For the salient point sets, a similarity function is required which encompasses missing points in in order to make the search occlusion invariant.
  In [21], it is proposed to define perceptual similarity rather than mathematical similarity. In [17] the use of color features by their perceptual impression is proposed. These are important extensions of the available similarity functions.
- After the query, the result is usually ranked in order of descending similarity. The query may be repeated with an image selected from the result set to achieve a form of visual browsing.
  The user interface can be made more intelligent [20] by relevance feedback.

For the PictoSeek system [18], on a 500 consumer object database the viewpoint invariant color feature set with an EXOR similarity function results in 98of a different picture of the same object. That is, different recordings at different camera recordings of one object result in identifying the images as identical. The recall rates appear to be robust against up to 60% occlusion of the object They are also robust to a change in viewpoint up to 75 degrees.

From a database point of view this is a marginal data set, but the point of demonstration was in the recall rate not (yet) in scalability.

These are encouraging results, but there is a snag in these and almost all other reported figures in literature in the fact that many depends on the composition of the database and the suitability of the feature set for that specific domain and query. In fact, theses 1 and 2 in the section above guarantee that objective performance evaluation of image databases is an art on its own for which a simple standard solution does not exist.

2.3 Computational support for image archives

A large collection of computational methods has been developed dealing with several dimensions of the problem at hand.

A schematic overview is given in Table 2.3 where *image* stands for an

image processing	*image*		→	*scalarimage*	enhancement
image processing	*image*		→	*vectorimage*	gradient
image processing	*image*		→	*vectorsequence*	flow
segmentation	*image*		→	*object*	strong segmentation
segmentation	*image*		→	*patch*	weak segmentation
feature extraction	*image*	*object*	→	*vector*	color
feature extraction	*image*	*patch*	→	*vector*	partial colors
feature histogram	*image*	*object*	→	*histogram*	aspects
feature histogram	*image*	*patch*	→	*histogram*	partial aspects
model estimation	*image*	*object*	→	*vector*	model match

Table 1 Classification of recurring computer vision functions (limited list).

image type scheme. An *image* can be one valued for gray valued images, or three valued for the various color space representations. It can also be scalar for intensity and color values, or vector fields for the gradient in the image and the motion pattern of a time sequence. As defined in the paragraph on segmentation, an *object* represents an indicator field indicating which pixels constitute the object. A *patch* is a (point-wise) subset of *object*. Multiple feature values in a *vector* of one image or one *object* or *patch* are captured in a *histogram*. These types often take the form of a type hierarchy in a programming language such as C++ or, equivalently, a class hierarchy in an O-O language.

From the list of operations, it is concluded for purpose of image databases that the set of data structures for computer vision is large, much larger than for common database applications.

After the preprocessing and segmentation steps selected from Table 2.3, features are computed which serve as an index to the image content. Table 2.3 contains a list of features with a classification of their invariance.

An index to the image data base differs significantly from the index in common databases. An index in a standard database is based on one or more attributes taking values from rather simple domains, sorted into a (multi-dimensional) search structure, or transformed and organized as a hash data structure to speed up retrieval. Indexing images is different in the following ways. First, the indexing information is always derived from the underlying image rather than a pure copy of part of the image representation. Second, only the index is used to answer a query, the underlying data is not considered due to its excessive size and processing requirements. Finally, the image carries a visual representation and often involves several objects of interest. This implies that the index should also work under a subset of the invariant features included in its encoding. Only at the very bottom-end of the query, the original data may be used in a model match approach to fit the specific pose of the query object to the data to assure the query has yielded the desired result.

type of index	data in	data out	example	ill. inv.	vwp. inv.	segment. required
global	*image*	*histogram*	average R,G,B	-	-	
global	*image*	*vector*	average hue	-	-	
object	*objects*	*vector*	size	+	-	strong
object	*patches*	*vector*	texture	+/-	+/-	weak
object	*patches*	*vector*	color R,G,B	-	-	weak
object	*patches*	*vector*	color l_1, l_2, l_3	-	+	weak
object	*patches*	*vector*	color m_1, m_2, m_3	+	+	weak
object	*object*	*histogram*	color	+/-	+/-	weak
object	*patches*	*histogram*	color	+/-	+/-	weak
object	*object*	*pointset*	geom. hash	+/-	+	weak
object	*object*	*model*	geom. model	+/-	+	strong

Table 2 Classification of some recurring indexing functions (only two invariance classes indicated).

This operation is so costly and the variety of poses is so big, that it stresses the role of a database index as a filter to its extreme.

Given the complexity and size of the image database, the index is often an internal measure aimed to support specific class of queries. Several indices may be needed to obtain the required query selectivity and to support a broad class of queries.

When running the query, the query representation is compared with the index by the similarity function as listed in Table 2.3.

type of index	data	circumscription of similarity match
global image characteristics	*histogram*	histogram matching
global image characteristics	*vector*	proximity match
global image specifics	*vector*	presence / absence
object characteristics	*vector*	presence / absence
object configuration	*histogram*	histogram match / presence
object configuration	*pointset*	affine invariant point set match
object match	*vector*	geometric model matching

Table 3 Classification of some recurring similarity functions.

The last issue to consider is the definition of the query. From a database perspective this is very simple. The query consists of just one example image (to search a similar one in the database). Alternatively it consists of a sketch of patches with a desired spatial relationship.

The research question at this point is when similarity rather than exact retrieval is essential for image databases, how to design query optimization procedures for mathematical query functions rather than the logical query comparisons in common databases.

3 DATABASE SUPPORT FOR IMAGE RETRIEVAL

Research in database management has reached a state where relational database systems are readily available to manage large amounts of data. The pervasive use and effectiveness of a DBMS can be attributed to the following:

- *A concise data model*, which provides a high-level abstraction of the data items, their relationships, and their properties using a closed mathematical framework. As a consequence a strong asset of a database that it can be kept in a known state. All admissible input states are known a priori. This helps to ensure integrity.
- *A calculus and algebraic query language*, which provides for a computational complete framework to retrieve and manipulate database portions without concern about their algorithmic behavior.
- *Physical independence*, which permits a user to ignore the (physical) storage layout and the details of the algorithmic layers for its maintenance. The mapping from a query expressed against the logical data model is automatically compiled into the most efficient storage realization. The availability of such conceptual layers provide the means to optimize storage and querying at levels of abstraction. There is no need to reconsider the design of the entire database system when concentrating on solving complex queries. The modularity has been an important factor in the development of the database as an established field.
- *Closed world assumption.* When the database is in a guaranteed state, and the query is within the list of admissible queries, a negative search result will carry important information: the requested is not contained in the database. The trustworthiness of such a negative answer has high significance, which cannot be easily guaranteed for sensory data indexed using the feature sets. As the image data are projected on sets of features (a non reversible reduction of the image data), the object may be undetectable by the indexing features while it still is in the database somewhere.

Despite the many relational and object-oriented database management systems produced over the last two decades, no system has been produced that solves all data management problems incurred. The DB community estimates that at least 80% of all data still resides outside the confines of a DBMS, i.e. in bulk stores (audio, video, images) and files (word processing, consumer use).

The difficulty in extending the DB-technology to these new sensory data is threefold: 1) the size and storage structures of these new data types, 2) the specific type and language of query which come along with audio, video, images (and free text), and 3) the access to the content of these items.

Items 1) and 2) in this list are within or close to the DB-paradigm. Current technology enables a user to introduce new atomic types together with its operators that suit the application. Commercial systems such as Oracle and Informix already provide a step in this direction using their DataCartridge and DataBlade technology, respectively. They provide libraries of data structures with operations to support geometrical algorithms, for GIS applications, and image manipulation. Unfortunately, the extension modules provided are just a first-generation solution. The modules have to be defined by someone fluent in database technology, because interaction with the various database components is tricky. Full use of the DBMS is limited as well, because the query optimizers are generally not equipped to exploit the properties of the user defined enhancements.

The item 3) is well outside the current DB-paradigm. Sensory and text items do not permit an algebraic query set. The natural way to approach them is by example, a set of positive and negative examples, or other means of association. They also do not provide a concise data model. The variety and the interpretation of the content cannot be separated from the question. As mentioned above, images come with a multitude of interpretations. Often, only portions identified with a segmentation algorithm carry properties for retrieval. Where the database technology in its development has profited from the closed search space spanned by the data dictionary and the query set, such search spaces are not clearly defined for sensory (and free text) domains.

Construction of a sizeable image archive with its complementary query language is still an open research issue. Problems faced are:

- *type scheme*, dealing with images requires a rich typing scheme surpassing the capabilities of object-relational systems
- *computational scheme*, query processing calls for proximity queries rather than yes/no decisions on objects under consideration.

This leads to overstressing the capabilities of a standard database schema. In a relational system each object is cast into a tuple, i.e. a fixed and limited set of attributes relevant for a large collection of similar objects. Alternatively, the information is encoded into a relational table and let the user interpret the results. The situation in object-oriented systems is not much better. Although they provide for a richer data model, it is not possible to partially include an object into the hierarchy or to let an object participate in multiple classes at the same time.

But even if we restrict our image archive to those cases where we can describe the properties in a database table (or class hierarchy), the computational model underlying a query language interpreter is too strict. In the DBMS field, a query predicate can be evaluated against a database with absolute precision. The predicate holds or is false. There is no middle way, nor a ranking scheme. Given incompleteness of the information available to classify an object this computational model is bound to fail. The least that is needed in the handling of sensory data is: 1) a proximity-based computational model where the DBMS returns the answers together with a value between 0 and 1 to indicate confidence that the query predicate holds, 2) index and parameter domains which are continuous to ensure enough selective precision. If sensory data are categorized in qualifiers "large", "small", "yellow" rather than the physical measures not only the discriminatory power is lost to access 100,000+ databases, but also such qualifiers have no meaning without the question. "Yellow" is only "yellow" if "orange" or "ocher" is excluded from the query.

4 SYSTEMS UNDER DEVELOPMENT

In the large scale AMIS-project coordinated by the University of Amsterdam with participation of the CWI's database group, University of Utrecht's spatial data structures and the University of Twente with Quality of Service management as well as query optimizations. The envisioned architecture for experimentation consists of three layers of activity: storage and WWW access, query and feature detectors, and application layer.

The storage layer is build around the Monet extensible database system maintained at CWI. It provides access to both multi-media data stored locally and accessible through their URLs, and the multi-media archive maintained in the form of a large collection of CD-ROM's at University of Twente.

The input side is dealt with using a feature detector engine, which uses black & white box feature detectors to derive static feature values (vectors) from multi-media objects. Their results are kept around in feature indices for query support. The output side deals with effective support for querying the multi-media database. Both in terms of ranked responses using the feature indices and by exploitation of the inner structure of Web pages.

The top layer contains a number of applications to highlight and exploit various aspects of the platform. A simple search-engine like interface is provided to gain direct access using selections on the features maintained. An administration interface provides access to the database internals and statistics. A simple data entry form can be used to register new feature detectors and sources of information to index as soon as possible. University of Amsterdam's PictoSeek provides an engine to search the database using image filters and characteristics. The PictoSeek system can be viewed at

http://www.wins.uva.nl/research/isis/PicToSeek.

Next to the AMIS-project, in the companion digital media warehouse project, the middle layer comprises two components, roughly dealing with input and output. In this project, the collection of search techniques is extended with demonstrator video and audio service demos, as well as novel ways to query semi-structured data, e.g. the Web at http://www.cwi.nl/ acoi.

5 CROSSING THE DIVIDE

To make true progress in image databases requires a clear delineation of the research problems. The following issues form the core.

1. Discrimination on the basis of image content is the sole means to locate images of interest.
2. The expressiveness of a visual imprint of an object cannot be captured in verbal or categorical expressions. As a consequence, the query should be specified in visual means.
3. A critical issue is the definition of useful image features to serve as an index, or rather a range of features expressed in the same data structures. In the paper we have made an attempt to order some of the features by classes of use and data types. This is the best opportunity to make contact to expand the connection further into the database paradigm.
4. An image may have be present in the archive in 1 of a 1,000,000 different incarnations. Features, indices as well as similarity functions have to be able to deal with that. This is the prime research question in image databases as it implies a new view on indices and similarity measures, as well as a new view on image retrieval algorithms. For example, the fact that sensory similarity measures are a mathematical rather than a logical formalism requires new solutions.
5. To keep the query specific for not only the query object, but also the desired classes of invariance determine the implementation of the actual search. Again, a new research topic posing research questions for query optimization for a wide variety of invariances, i.e. query classes. In the current practice, the topic is underrated and usually solved by a system specific choice for one set of queries suited for one domain, e.g. art or trademarks.
6. Indexing an archive is never complete due to the open-ended list of possible queries formulated after the archive was defined. Closed world solutions do not last long for an archive (any archive whether text or pictorial). This requires dynamic solutions for the data dictionaries, search strategies as well as index optimizations.
7. Similarity retrieval is an essential part of dealing with sensory data, for which new ways of query optimization are needed.
8. Another essential element of sensory data is the handling of incomplete information. Part of the object may be out of sight while its presence should be detected still.
9. And, finally, when image databases work, the question how to integrate with other modalities such as free text, categorical information and sound returns on the agenda.

REFERENCES

[1] Grosky W. and Mehrotra R., Special Issue on Image Database Management, Computer, Vol. 22, No. 12, 1989.

[2] IFIP, Visual Database Systems I and II, Elsevier Science Publishers, North-Holland, 1989 and 1992.

[3] Image Databases and Multi-Media Search, (eds. A.W.M. Smeulders and R. Jain), Series on Software Engineering and Knowledge Engineering, Vol. 8, World Scientific, ISBN 981-02-3327-2, 1997.

[4] Jain, R., NSF Workshop on Visual Information Management Systems, SIGmod Record, Vol. 22, No. 3, pp. 57-75, 1993.

[5] Levkowitz, H. and Herman G. T., GLHS: A Generalized Lightness, Hue, and Saturation Color Model, CVGIP: Graphical Models and Image Processing, Vol. 55, No. 4, pp. 271-285, 1993.

[6] Ogle, V. E. and Stonebraker, M., Chabot: Retrieval from a Relational Database of Images, IEEE Computer, Vol. 28, No. 9, 1995.

[7] Sclaroff, S., Taycher, L., La Cascia, M., ImageRover: A Content-based Image Browser for the World Wide Web, In: Proceedings of IEEE Workshop on Content-based Access and Video Libraries, CVPR, 1997.

[8] Frankel C., Swain M. and Athitsos Webseer: An Image Search Engine for the World Wide Web, TR-95-010, Boston University, 1995.

[9] Shafer, S. A., Using Color to Separate Reflection Components, COLOR Res. Appl., 10(4), pp 210-218, 1985.

[10] Proceedings of Storage and Retrieval for Image and Video Databases I, II, and III, Vol. 1,908; 2.185; and 2,420; W. Niblack and R. Jain, (eds.), SPIE, Bellingham, 1993, 1994 and 1995.

[11] Proceedings of Visual Information Systems: The First International Conference on Visual Information Systems, Melbourne, Victoria, Australia, 1996.

[12] Proceedings of Visual Information Systems: The Second International Conference on Visual Information Systems, San Diego, USA, 1997.

[13] A. del Bimbo, M. Mugnaini, P. Pala, F. Turco: PICASSO: visual querying by color perceptive regions. In: Proceedings of Visual Information Systems, San Diego, USA, 1997, page 125 - 131.

[14] William I. Grosky: Managing Multimedia Information in Database Systems. CACM 40(12): 72-80 (1997)

[15] Amarnath Gupta, Simone Santini, Ramesh Jain: In Search of Information in Visual Media. CACM 40(12): 34-42, (1997)

[16] Pentland, A., Picard, R. W. and Sclaroff, S., Photobook: Tools for Content-based Manipulation of Image Databases, International Journal of Computer Vision, 18(3), pp. 233-254, 1996.

[17] A. del Bimbo, M. Mugnaini, P. Pala, F. Turco, L. Verzucoli: Image retrieval by color regions. In: Image Analysis and Processing, Springer

Verlag 1131, page 180 - 185.

[18] Gevers, T. and Smeulders, A.W.M., PicToSeek: A Content-based Image Search Engine for the World Wide Web, Proceedings of Visual Information Systems, San Diego, USA, 1997, page 93 - 100.

[19] Flickner, M. et al, Query by Image and Video Content: the QBIC system, IEEE Computer, 28(9), 1995.

[20] R. Schettini, A. Della Ventura, M. T. Artese: Color specification by visual interaction. The visual Computer vol 9-6, 143 - 150, 1992.

[21] S.Santini, R. Jain: Visual navigation in perceptual databases. In: Proceedings of Visual Information Systems, San Diego, USA, 1997, pages 101 - 108.

[22] Smith, J. R. and Chang S.-F., VisualSEEK: A Fully Automated Content-based Image Query System, In Proceedings of ACM Multimedia, 1996.

BIBLIOGRAPHY

Arnold W.M. Smeulders is professor of Computer Science and director of the Computer Science Institute. His chair is in multi media information processing. He leads the *intelligent sensory information systems* research group with Martin Kersten. The main research attention of the ISIS-group are in computer vision for industrial applications, (video) document access, theoretical foundations in vision in particular mathematical morphology, color image analysis, performance and evaluation in vision and image databases. He is associated editor of IEEE transactions PAMI, chairman of the Dutch chapter and co-chair of the TC on multi- media of the International Association of Pattern Recognition, European program chair of IEEE Multimedia 99 to be held in Florence and chair of the Visual Information Systems III to be held in Amsterdam, June 99.

Martin Kersten received his Ph.D. degree in computer science from the Vrije Universiteit, Amsterdam in 1985. After his graduation he moved to CWI, the national research center for mathematics and computer science in the Netherlands, were he established the database research group. Since 1993 he has developed the Monet extensible database management system and deployed it in several novel application domains, such as GIS, multi-media, and data mining. In 1995 he co-founded the company Data Distilleries to commercialize the data mining technology. He is currently head of the Information Systems department of CWI and full professor at the University of Amsterdam. His research interests are: distributed and parallel database system architectures, their performance, data mining, and multi-media database applications. He is (co-)author of over 120 technical papers, associated editor of VLDB journal and Kluwer Distributed and Parallel Databases, and active reviewer of European Community projects.

Theo Gevers received his Ph.D. degree in Computer Science from the University of Amsterdam in 1996 for a thesis on color image segmentation and retrieval. His main research interests are in the fundamentals of image database system design, image retrieval by content, theoretical foundation of geometric and photometric invariance and color image processing.