Chapter VII

Design and Evaluation of a Content-Based Image Retrieval System

D. McG Squire
Vision Group – CUI – University of Geneva
Monash University – Melbourne, Autralia

H. Müller, W. Müller, S. Marchand-Maillet and T. Pun
Vision Group – CUI – University of Geneva

The growth in size and accessibility of multimedia databases have changed our approach to information retrieval. Classical text-based systems show their limitations in the context of multimedia retrieval. In this chapter, we address the problem of conceiving and evaluating a content-based image retrieval system. First, we investigate the use of the query-by-example (QBE) paradigm as a base paradigm for the development of a content-based image retrieval system (CBIRS). We show that it should be considered as a complement to the classical textual-based paradigms. We then evaluate the capabilities of the most up-to-date computer vision techniques in contributing to the realisation of such a system. Further, beyond the necessity of accurate image understanding techniques, we show that the amount of data involved in the process of describing image content should also be considered as an important issue. This aspect of our study is largely based on the experience acquired by the text retrieval (TR) community, which we adapt to the context of CBIR. Similarly, the text retrieval community has also developed a significant experience in evaluating retrieval systems, where judgements include subjectivity and context dependency. Extending this experience, we study a coherent framework for performing the evaluation of a CBIRS.

As a practical example, we use our Viper CBIR system, using a novel communication protocol called MRML (Multimedia Retrieval Markup Language) to pinpoint the importance of the sharing of resources in facilitating the evaluation and therefore the development of CBIRS.

INTRODUCTION

As more and more multimedia data and databases are accessible from the World Wide Web (WWW), it is fundamental to design tools which allow for the efficient browsing of such repositories. We address here the problem of the design and evaluation of a content-based image search engine. The conception of such a system can be approached from a number of viewpoints, ranging from computer vision to software engineering.

A number of image search engines are currently available on the WWW, either as commercial products or as research demonstration prototypes and are described in the related research literature. While they differ in their approaches at various levels, they all have the common goal of providing assistance to a user for retrieving a visual document within a database. In order to ensure the usefulness and usability of such a system one should look carefully at the system from the user viewpoint and determine what level of assistance the system should provide to different types of users. This in turn allows one to define which technique should be used for formulating image queries. Computer vision procedures remain the core of a content-based image retrieval system (CBIRS) since they provide the representation under which the documents will be compared with one another. While large advances in image analysis techniques have been made, automated image content understanding processes still misses a substantial part of the semantic content of a visual document. This strongly influences the way in which images will be represented internally and therefore the choice of the features that should be extracted. This choice is also inherently very closely related to the internal search strategy taken. The background of database management should fully be exploited here. Finally, the architecture under which all the various components that form a complete CBIRS will communicate should be studied closely in the context of software engineering, in order to take full advantage of the context in which these tools will be developed (e.g. WWW).

In this chapter, we investigate all aspects of a CBIRS, from the usability to its technical development and evaluation. This investigation is done in close relationship with the experience acquired from the text retrieval (TR), which is a mature field when compared with that of image retrieval. We study the problem of conceiving a CBIRS from different viewpoints. Typically, a trade-off is to be defined for satisfying the constraints arising from the user, the development and the architecture of the system. Then, we advocate the construction of a consistent framework for the evaluation of all existing CBIRS. One key point in this framework is the sharing of resources, from a common database associated with validated ground truth to software components allowing one to distinguish between the evaluation of the different characteristics of a system such a retrieval efficiency and usability and to share experimental knowledge. Here, ground truth is understood as the given of real user judgements on image similarities. Another crucial factor is the derivation of a coherent set of quantitative measures so that objective comparison can be performed between the different approaches taken and the assumptions made.

Further, we use the practical example of the *Viper* system which has been under development for several years in our group, to illustrate the construction of a CBIRS and to introduce a novel approach for distributing and sharing multimedia resources and software components via the definition of a communication protocol called MRML (Multimedia Retrieval Markup Language). Practical results are provided to illustrate the evaluation context we recommend. We conclude by identifying the problems that still prevent CBIRSs from being fully effective and sketch some ways of overcoming such deficiencies.

CONTENT-BASED QUERYING OF IMAGES

To make the access to the data simple, it is very important that the data request may be formulated in a way that corresponds as much as possible to the user's intuition. Typical search engines use a text-based query paradigm where the user is asked to enter keywords or some free text that best describes the information he or she is looking for (*e.g.* (AltaVista, 2000)). Such a query system calls for the existence of corresponding keywords associated with each data in the database. Moreover, this querying paradigm assumes that the data in question can be described by a small set of keywords or phrases. This may not be the case when the information sought consists of visual information such as that contained in an image. Although the global content of an image may easily be described by the names of the objects composing it, it may be more subtle to exactly describe the «flavour» of the image content. This may even be more true when the user only has a vague idea of what he is looking for.

The query-by-example (QBE) paradigm comes as a solution to the above problem. In this section, we introduce the features of a typical QBE-based CBIR system. The analysis is made from various viewpoints in order to better evaluate the implications and assumptions of such a system.

Search paradigms and interactivity

Usual querying systems use textual keywords for describing the information sought. The advent of multimedia databases, and particularly image databases, has shown that this type of querying may be too restrictive in the context of multimedia retrieval. This is due to the fact that textual annotations may not permit enough flexibility to easily describe the content of an image or to formulate a query that encompasses all aspects of the user's needs.

The QBE paradigm proposes to adapt the query formulation to the particular type of data under investigation, namely the images. The foundations of the QBE paradigm can be formulated as follows.

- A user finds it easier to show examples of what he is looking for than actually describing it,
- The query is more compact and natural for the user,
- The results can be evaluated by the user through direct comparison with the initial query.

Ideally, the user should show an example and the CBIRS would return all images in the database that correspond to this query. Most of the existing systems allow for getting a random sample set of images in their database, so as to provide the user with a starting point. Since most of these systems mostly aim at demonstrating their capabilities in retrieving similar images, the user is not given full flexibility for the choice of the starting point. A possible solution to this problem is to allow the user to *create* is own starting point (sketch or image composition). However, allowing the user to submit images from his own as a starting point requires to include somehow these images in the database for comparison. Since this may involve a long calculation time, this feature is not always available (see next section for details on how similarity is generally understood from the computer viewpoint).

In any case, it is unlikely that the user will find an example image containing *exactly* all features needed (if he were to find it, he would not look for an other such example). Therefore, it is natural to think of an extension of the QBE paradigm as a querying system where the user would show *multiple* examples and the CBIRS would gather all the information presented into a *pseudo*-image and respond with the best possible match from the database. In this respect, it should also be noted that the QBE paradigm comes as a *complement* of the text-based query paradigm, rather than as its *replacement* (QBIC, 1998).

Due to the user subjectivity and also the non-completeness of the database in providing examples (ie the non-continuity of the search space), it is not realistic to assume that the CBIRS will respond with the correct answer at the first querying step. The key here is therefore to provide the user with a mean of *improving* the result on the basis of a previous search. One way is to exploit relevance feedback as a generalisation of QBE. The returned set of images serves as a basis for re-formulating the query by marking positive and negative matches, so as to emphasise or disregard information in subsequent search passes. This will therefore create a dynamic dialog between the user and the CBIRS, simulating on-line learning and therefore a form of target search (Müller *et al.*, 1999a), (Müller *et al.*, 2000a).

Image representation

The major aim of the QBE is therefore to place the user in a more comfortable situation by making the querying system more intuitive. From the computer viewpoint, things are however different. By definition, the QBE paradigm relies on a thorough analysis of the content of the query image. This type of paradigm therefore requires to use all capabilities of computer vision techniques for such a image understanding task. It is not our aim here to describe a complete solution for this part of the system but rather to analyse and comment alternatives which are typically found in the computer vision literature.

The analysis of an image content is done via the definition of image characteristics whose representation are stored into a vector, the *feature vector*, on which subsequent similarity analysis may be performed. Image characteristics range from the analysis of colour to the definition of salient points such as corners. Higher level characteristics such as object shape or local curvature can also be used to characterise the content of an image. Specific application domains may call for the definition of highly specialised features from representing the image content. This is the case for example in medical imaging, where images of brain MRI look extremely similar to the novice user. All these features may apply to the complete image or to a partition element of the image via quadtree or any other partitioning technique. There are also different ways of storing these features for subsequent usage. Some of the image features may be summarised in a histogram. A typical example is the colour histogram of an image, which summarises the occurrence of every possible colour (with respect to a quantisation scale) in the whole image. One advantage of such a representation is that it captures the complete image at once and is therefore inherently invariant under rotation. It may therefore be of use in some applications. However, the global nature of this characteristic does not allow to describe precisely the image content. One solution is therefore to proceed with histograms on a recursive partitioning of the image. Each image part then is associated with its own characteristics. In any case, a trade-off is to be made between the invariance (and globality) and its specificity (and locality) of the image representation.

Colour features: Colour is the first feature one can think of for characterising the content of an image. The definition of the colour space, in which any colour value will be represented allows for emancipating from constraints such as illumination and therefore object viewpoints. The definition of the most common colour spaces can be found in any computer vision reference book such as (Jain, 1989). A more thorough study of colour spaces designed for obtaining illumination invariance can be found in (Gevers and

Smeulders, 1996).

The design of colour spaces may be driven by different factors. Some colour spaces are related to the conception of devices (*e.g.* the CMYK model for colour printers). In our case, the most important colour spaces are the ones which are based on psycho-visual considerations. This is the case, for example, for the HLS and HSV models. In these colour spaces, the associated Euclidean distance is said to represent well colour similarity, as understood by the Human Visual System. The YUV colour space is also often used since it as been defined as the standard colour space used for the MPEG video compression norm.

Textural features: Texture has been widely recognised as a crucial characteristic for completing the analysis of an image content. Texture can only be understood via the definition of a scale, which itself defines a neighbourhood size. Hence, the analysis of texture is typically done using either filters or statistical parameters.

Using filters such as Gabor filters (Ma and Manjunath, 1996), (Jain and Healey, 1998) at different orientations and scales allow to define global texture parameters. The response of a bank of filters covering the frequency and orientation spaces associates each texture location with a vector of characteristics. The study of the statistical properties of the distribution of such characteristics should allow for uniquely characterising the texture with respect to its type (*e.g.* coarse, fine, coherent or otherwise).

An alternative approach is to characterise a portion of texture by a statistical model such as a Markov Random field (MRF) or using Hidden Markov models (HMM) (Rabiner and Huang, 1993). The parameters of the texture model thus obtained concisely characterise the texture and possibly allow for its re-generation.

Curvature: Recent image analysis techniques often see the image as a scalar field or, more generally, as a surface (Lindeberg, 1994). Using differential geometry, the study of the properties of this surface has provided new characteristics for describing the image content (Schmid and Mohr, 1997), (Winter and Nastar, 1999). Surface curvature is one of these properties. It is typically understood as the curvature of the level sets associated with the surface (*mean curvature*). Again, the properties of the (local or global) distribution of the curvature may be used to characterise the content of an image.

High level features: For specific tasks, it may be useful to develop highly specialised features, based on some a priori knowledge. Such high level features range from the definition of shape descriptors to filters, which allow for the localisation of specific image parts.

Depending on the type of data involved, shape may be such a criterion. The objects contained in an image should first be isolated via some form of segmentation. Segmentation techniques range from simple thresholding or edge detection to more adaptive methods such as snakes (Kass *et al.*, 1987). The objects are then considered separately and features may be obtained from their characteristics or neighbouring relationships. Contour-based shape descriptors map the contours of the objects present in an image onto some compact description (*e.g.* length, minimum or maximum curvature, area). It may however not always be possible to define uniquely such contours. Skeleton-based shape analysis comes as a solution to this problem. Binary shape skeletons have been studied for long and successfully applied in different domains. The extension of these technique to grey-scale or colour image analysis is more recent but also very promising (Wieckert, 1998), (Sethian, 1999). By defining such central structures, the objects present in an image can be described in terms of their topology and neighbouring relationships without the need for specifically isolating them. The skeleton (also called medial axis) can then be summarised into a concise description by storing some of their characteristics (*e.g.* junction and end points, stroke lengths and curvatures).

Finally, characteristics arising from supervised learning (*i.e.* model parameters learnt through examples) may be useful in specific applications. This is the case for example for human face location (Rowley *et al.*, 1998). This process has generated a lot of interest from forensic and security application developers. Typically, models such as Neural Nets (NN) or MRF are trained with face snapshots. The model then becomes a filter and it is the response of an image portion which determines the probability for a specific object (*e.g.* a human face in this case) to be present at this location.

Similarity measures: The unification of concepts such as colour and texture into a feature vector allows for a quantitative evaluation of similarity. Generally, all the normalisation effort is done in the design of the features. Therefore, classical distance measures like the Euclidean distance can be used in the corresponding n-dimensional space thus defined. An extension of this distance is the Mahalanobis distance, which takes into accounts the variance of the data in each direction of the feature space. More elaborated weighted distance measures may be designed so as to model closely the user notion of similarity (Zobel and Moffat, 1998).

Data management

Beyond the problem of finding a suitable representation for searching for an image is the problem of managing the amount of data in a efficient way. Typically, feature vectors span a n-dimensional search space which should be represented efficiently so as to promptly solve nearest neighbour problems. All this actually goes down to utilising efficient database management techniques.

CBIR is a fairly young field when compared to other IR fields where much work has already been done. This is the case in the fields of speech recognition and text retrieval where similar problems arise. The data is to be accessed often and quickly and therefore this imposes to look carefully at the data organisation.

Many authors propose methods for reducing the time taken to search the feature space, such as dimensionality reduction using Principal Components Analysis (PCA) (Pun and Squire, 1996), (Sclaroff *et al.*, 1997), clustering (Jain and Vailaya, 1996), (Vellaikal and Kuo, 1998), or spatial search structures such as R-trees (Güting, 1994), KD-trees (White and Jain, 1996), (Sclaroff *et al.*, 1997). The suitability of PCA as preprocessing for information retrieval has been challenged since it can eliminate the «rare» feature variations which can be very useful for creating a specific query. The other techniques all limit search by pruning the number of images for which distances are calculated (Squire *et al.*, 1999) or by limiting the number of features calculated (Müller *et al.*, 2000).

TR researchers, however, have more than 30 years of experience with the problem of reducing query evaluation time in text retrieval systems. They have generally taken a different approach, based on the *inverted file* (IF) data structure most commonly used in TR. An IF contains an entry for each feature consisting of a list of the items which contain that feature. In general, an item has only a small subset of all possible features: similarity computation is then restricted to the subspace thus defined. Several IF-based search pruning methods are described in (Witten *et al.*, 1994). The principal difference is that in IF systems similarities are evaluated feature by feature, rather than item (document, image) by item. This search may be pruned in two basic ways:

- do not evaluate all features present in the query,
- do not evaluate all features for all possible response documents.

The choice of which features to evaluate depends on their importance, typically defined by a relevant weighting scheme (see below for an example).

System components and organisation

As mentioned above, the design of a CBIRS does not only involve the issues of defining a good querying system and a suitable image representation. A CBIRS is aimed at managing a substantial amount of data and, in this respect should be treated as a complex system. We advocate the use of the latest software engineering techniques in order to obtain the best possible system component organisation. We sketch here the main remarks that may be made regarding this aspect of the design of a CBIRS.

The above sections made clearly apparent that the structure of any CBIRS may be divided into three parts, namely, the interface, the search engine, and the indexing engine. Each of these parts should be analysed as an independent component so as to allow its exchangeability and re-use.

User interface: The user interface represents the visible part of the CBIRS. This software piece should therefore be realised using at best the Human-Computer Interaction knowledge. This is clearly not the case at present. One major reason for this is that in current systems, the interface is integrated into the system and is therefore developed by the same people that developed the rest of the system.

We strongly believe that the usability of the interface is an important factor in the quality of the global system. Even if the rest of the system offers powerful features, if these features are not well-presented in the visible part of the system, the user may miss these features, simply because it takes too much expertise to use them.

Image search engine: The core of the CBIRS is the image search engine. It is this part of the system which will process the user query and respond accordingly. The main task of this part of the system is to retrieve stored information and to solve nearest-neighbour problems, based on the similarity measures. Typically, the online processing should be made small so as to respond as quickly as possible, in order not to lose the user's attention. In this respect, it is crucial to transfer as much processing as possible to the pre-processing part, namely, the indexing engine.

Image indexing engine: This part of the system aims at preparing the data so that it can be retrieved in the fastest possible way. It is the indexing engine which will perform all the computer vision tasks, which is the part supposed to be the most computationally expensive. However, this is valid only in a system where the user is asked to chose examples from the database and not submit his own. One important characteristic of this part of the system is that it should be conceived for being incremental. The construction of the indexing representation, be it a matrix of inter-distances or an inverted file should be based on the completion of the structure rather than it re-calculation.

Architecture: From the above, it is clear that some form of independence is to be ensured in the development of these three software components. By nature, the indexing engine is fairly independent from the rest of the system. It will typically create a database structure which will subsequently be used by the search engine.

The search engine and the interface should communicate to allow the user queries to be processed. We strongly advocate the use of a communication protocol which will ensure this independence (see the section on MRML, later). Further, the image search engine should support the largest part of the online computations. The rapidity in which the system responds is recognised as an important factor for its quality (Nielsen, 1993). Above a certain response time, the user attention is lost and the interactivity cannot be kept consistent. It is therefore important to speed up the search in every possible way. Well-known search techniques exist to split the search into sub-parts and then merge results. The design of an image search engine may therefore benefit greatly from a distributed architecture with protocols such as CORBA.

EVALUATION OF A CBIRS

The current status of performance evaluation in CBIR is far from that in TR. Many different groups are working with different sets of specialised images. There is neither a common image collection, nor a common way to get relevance judgements, nor a common evaluation scheme. In this section, we address the problem of creating a context in which a CBIRS can be evaluated objectively. We base our study on the example of the Text Retrieval community, in which progress have been made in this direction and where standards are already defined. We then study how these results could apply to the context of CBIR. Based on the experience developed in our group (Müller *et al.*, 1999), we conclude by proposing various methods for the evaluation of a CBIRS.

The example of the TR community

In the 1950s, TR researchers were already discussing performance evaluation, and the first concrete steps were taken with the development of the SMART system in 1961 (Salton, 1971a). Other important steps towards common performance measures were made with the Cranfield test (Cleverdon *et al.*, 1966). Finally, the Text REtrieval Conference (TREC, 1999) series started in 1992, combining many efforts to provide common performance tests. Co-sponsored by the National Institute of Standards and Technology (NIST) and the Defence Advanced Research Projects Agency (DARPA), TREC has been held annually since its inception – 2000 will see TREC-9. The TREC project provides a focus for these activities and is the world-wide standard in TR (Vorhees and Harmann, 1998). Nevertheless, much research remains to be done on the evaluation of interactive systems and the inclusion of the user into the query process. Such novelties are included in TREC regularly, *e.g.* the interactive track in 1994.

Salton (Salton, 1992) gives an overview of TR system evaluation. Although performance evaluation in TR started in the 1950s, here we focus on newer results and especially on TREC and its achievements in the TR community. Not only did TREC provide an evaluation scheme accepted world-wide, but it also brought academic and commercial developers together and thus created a new dynamic for the field. For example, in 1999, sixty-six groups representing sixteen countries participated in TREC-8 and it is assessed that retrieval system effectiveness has approximately doubled since TREC-1 in 1992.

Data collections: The TREC collection is the main collection used in TR. At present, TREC participants must index a collection of about 2 Gigabytes of textual data at the conference itself. Comparisons of participating systems are given later. A large amount of training data is also provided before the conference. Different collections exist for different topics, and several evaluation methods are used. Special evaluations exist for interactive systems (Over, 1998), spoken language, high-precision and cross-language retrieval. The collections can grow as computing power increases, and as new research areas are added.

Relevance judgements: In order to make objective comparisons, well-defined relevance criteria have to be defined. This is however a non-trivial task, due to the subjectivity of the user in interpreting the data. In this respect, TREC has defined a strict context for determining if a document is relevant or not, with respect to a given query. In TREC, the relevance of a document can be defined as follows.

"*If you were writing a report on the subject of the topic and would use the information contained in the document in the report, then the document is relevant. Only binary judgements («relevant» or «not relevant») are made, and a document is judged relevant if any piece of it is relevant (regardless of how small the piece is in relation to the rest of the document).*"

The determination of relevant and non-relevant documents for a given query is one of the most important and time-consuming tasks. Using real users, it takes a long time to judge a large number of documents. Since it is unreasonable to expect humans to examine 2 Gb of data, a pooling technique is used for the TREC collection (Spark Jones and Rijsbergen, 1975). Only a subset of the collection, which is considered to be complete for a given query, is presented to users for actual relevance judgements.

All this leads to predefined user judgements, forming the ground truth in which user subjectivity is assumed to be thoroughly accounted for. This will form the base for subsequent results comparison and assessment.

Performance measures: Quantitative performance measures, which make use of this ground truth have now to be defined. Performance measures typically evaluate the distance between the above-defined user-based ground truth with the results provided by a given automated system.

The most common evaluation measures used in TR are *precision* and *recall* (Equation (1)), usually presented as a *precision vs. recall* graph (PR graph) (Salton, 1971), (Rijsbergen, 1979).

$$precision = P = \frac{\#\text{ of relevant documents retrieved}}{\text{Total }\#\text{ of documents retrieved}}$$

$$recall = R = \frac{\#\text{ of relevant documents retrieved}}{\text{Total }\#\text{ of relevant documents in the collection}} \quad (1)$$

Researchers are familiar with PR graphs and can extract information from them without interpretation problems. However, PR graphs may not contain all desired information (Salton, 1992) and several other measures are used at TREC, also based on *precision* and *recall.*

- $P(10)$, $P(30)$, $P(N_R)$ represent the *precision* after the first 10, 30, and N_R documents are retrieved, respectively and N_R is the number of relevant documents for the topic in question.
- The Mean Average Precision is the (non-interpolated) mean average *precision.*
- The *recall* at .5 *precision* is the *recall* at the rank where the *precision* drops below .5.
- $R(1000)$ represents the *recall* after 1,000 documents have been retrieved.
- The *Rank* of the first relevant represents the rank of the highest-ranked relevant document.

These key numbers offer a set of performance descriptors, so that different systems can be compared meaningfully and objectively.

We now will show how these results can map onto the field of CBIR. In particular, we advocate the use of common resources to create a suitable and evolving environment through which the evaluation of CBIRS will be performed.

Defining a common image collection

There are several problems which must be addressed in order to create a common image collection. The collection must be available free of charge and without copyright restrictions, so that images can be placed on the WWW and used in publications. The greatest problem is to create a collection with enough diversity to cater for the diverse, partly specialised domains in CBIR such as medical images, object images, face recognition and consumer photographs.

A common means of constructing an image collection is to use image databases such as the Corel photo CDs, each of which usually contains 100 broadly similar images (*e.g.* (Belongie *et al.*, 1998), (Ratan *et al.*, 1999), (COREL, 1999)). Unfortunately these images are copyrighted, and are not free of charge. Another problem in this case is that most research groups use only a subset of the collection, which can result in a collection consisting of several highly dissimilar groups of images, with relatively high intra-group similarity. This can lead to great apparent improvements in performance since it seems fairly easy to distinguish a tree from a car.

Another commonly used collection is VisTex, which contains texture images (VisTex, 1995). Although very useful for certain applications oriented towards textures, it is less useful for setting up a system where the aim is to distinguish between objects. A good candidate for a standard collection could be the images and videos from the MPEG-7 effort (MPEG, 1998), around which the international community has already gathered. Unfortunately, the collection is expensive and this data may not be shown on the WWW or in publications.

An alternative approach is for CBIR researchers to develop their own collection. Such a project is underway at the University of Washington in Seattle (Annotated DB, 1999). The collection is freely available without any copyright and has the advantage of offering annotated photographs of different regions and topics. It is still small with about 500 images, but several groups are contributing to enlarge the data set. The size of the collection should be sufficiently high that the trade-off between speed and accuracy can be evaluated. In TR it is quite normal to have millions of documents whereas in CBIR most system work with a few thousand images and some even with fewer than one hundred (*e.g.* (Müller and Rigoll, 1999)).

Defining a common software environment

In the above sections, the importance of the search paradigm in CBIR was demonstrated. Depending on the environment offered, the user will access the information sought more or less easily. In order to emancipate from usability constraints, it may also be important to define common software components which allow the user to have a consistent access to different CBIRS. This emphasises the need for studying carefully the architecture of a given CBIRS. Thanks to the above analysis, the three main components of a CBIRS were identified. This analysis shows clearly the independence of these three parts of the system. The user interface being the most visible part of the system, it should ideally be common to all CBIRS. This is largely not the case in current systems, which almost all have their own particular user interface.

The key to this problem is the separation of the user interface from the image search engine. We strongly advocate the construction of a standard multimedia communication protocol, which will allow to carry the queries between the interface and the image search engine. This way, compliant software components will be exchangeable and thus this will facilitate their evaluations. Such a communication protocol called MRML (Multimedia Retrieval Markup Language) is under development in our group and is detailed later in this article, in parallel with our retrieval system, *Viper.*

Evaluation based on user judgements

The evaluation of a CBIRS cannot be performed without referring to a ground truth. Ideally, this ground truth should be constructed according to user judgements of similarity. However, in CBIR there is not yet a common means of obtaining relevance judgements for queries. Even the inclusion of real users in the judgement process (as in TR) is not common. We analyse below the different techniques that are available to construct such a basis for the evaluation of a CBIRS.

Use of collections with predefined subsets: As already mentioned, a collection can be constructed from existing collections such as the Corel collection. Based on this, a very common technique to obtain relevance judgements is to exploit the existence of different topics in the collection. Relevance judgements are therefore given by the collection itself since it contains distinct groups of annotated images. However, the choice of sets can greatly influence the results, since some sets are visually distant from each other and others are visually closely related. Grouping is not always based on global visual similarity, but often on the semantics implied by the contained objects. Although consistent with reality, such a collection and its associated ground truth may therefore not allow for a pertinent evaluation of CBIRS.

Image grouping: A more general approach is for the collection creator or a domain expert to group images according to some criteria. The grouping is not necessarily based only on readily-perceptible visual features. Domain expert knowledge is very often used in medical CBIR (see (Shyu *et al.*, 1999), (Dy *et al.*, 1999), for example). This can be seen as real ground truth, because the images have a diagnosis certified by at least one medical doctor. However, it seems clear that an evaluation process based on such data is only relevant for CBIRS which aim at specialising in the corresponding domain. In their current state, generic CBIRS cannot have the deep understanding of the image content required in this context without being based on specialised features.

User judgements: The collection of real user judgements is time-consuming, but only the user knows what he or she expects as a query result. To obtain such judgements, relevance must be defined and the user must examine the entire database or a representative part of it (see TREC pooling (Spark Jones and Rijsbergen, 1975)). The user is then given a query image and asked to specify all relevant images in the collection. Experiments show that user judgements for the same image often differ (Squire *et al.*, 1997), (Squire *et al.*, 1999a), which is also observed in TR (Borgman, 1989). This is the only means of obtaining relevance judgements which acknowledge genuine differences between user responses, and does not assume the existence of one «optimal» query result. These individual differences are especially important if we want to demonstrate the ability of a system to adapt to the users' needs by using relevance feedback.

Simulating users: Some studies try to simulate users via an automated process. This is the case in (Vendrig *et al.*, 1999), where it is assumed that the user's image similarity judgements are modelled by the metric used in the CBIR system, plus noise. Such a assumption defeats the whole process of evaluation since, in this context, simulations can provide very good results – indeed, the quality of the results is controlled by the noise level.

Real users are generally very hard to model. It was shown in (Tversky, 1977) that human similarity judgements seem not to obey the requirements of a metric, and they are certainly user- and task-dependent. It seems therefore clear that fully-automated simulations cannot replace real user studies. Nevertheless, we believe there is room for research in the direction of *semi*-automating the acquisition of user judgements. This is motivated by two main reasons. First, getting user judgements is a time-consuming (and possibly costly) process. A semi-automated process may therefore help in deducing at maximum information for simplified and reduced user interaction. Secondly, it is not easy for a user to provide a consistent judgement during a long time. As described below, this process is tiring and the user may also change his opinion during the evaluation process (*e.g.* after seeing a large portion of the database, the user may adapt his notion of similarity). Hence, user judgements acquired over a long period of time cannot be blindly trusted. The number of images which a user must examine can be reduced by using pooling methods as in TR. However, the issue of changes of opinion still remains. There is therefore the need for developing tools that allow for validating the consistency of user judgements.

Summary: There are fundamental differences between the above methods. The ease of obtaining relevance judgements is an advantage of using collections with pre-defined groups of similar images. User judgements can still be made for such a collection. Domain expert knowledge should be used when it is available, such as in medicine and other specialised fields. For general CBIR tasks, we believe that the use of real users is essential (see (Squire *et al.*, 1997), (Markkula and Sormunen, 1998)). For a complete evaluation, the user with his/her expectations is a vital part of the system. It is essential that the user examines a significantly large fraction of the database, and that the relevance judgements are made in advance: users tend to be easily satisfied, even though the result may contain few, or even none, of the images selected as being relevant in advance. Further, the characteristics of the group of users from whom the relevant judgements are obtained are also very important. CBIR system developers of a system have different notions of image similarity from novice users.

Defining rigorous performance measures

Based on the ground truth, one may define different performance measures that will allow for quantitatively (and objectively) compare CBIRS. Before presenting such performance measures, we mention here two alternative which provide quantitative measurements without the need of having a predefined ground truth.

- User comparison: It is an interactive method where the user judges the success of a query as soon as the result is available. This technique considers the results as a whole and has the advantage of directly comparing the relationship between the results and the query. It is however hard to get a large number of such user comparisons as they are time-consuming and tiring. In fact, the user still needs to study carefully the internal structure of the answer (*e.g.* order and relevance of the images returned).
- Before-after comparison: To relax the above method, users are given two or more different results and are asked to choose the one they prefer or find to be most accurate. The comparison between two query results is more of a global process and the user can quickly issue an answer to each question. The bottleneck is that this method clearly requires a reference system for comparison. This technique may be mostly useful to validate the evolution of a given system throughout its development.

Single-valued measures: We now present quantitative measures based on the ground truth defined in the above section. These measures mostly derive from that used in the TR community.

- *Precision* and *recall* are standard measures in TR, which give a good indication of the system performance. Either factor alone contains insufficient information. For example, we can always make *recall* equal to 1, simply by retrieving all images. Similarly, *precision* can be kept high by retrieving only a few images. Thus *precision* and *recall* should either be used together (*e.g. precision* at .5 *recall*), or the number of images retrieved should be specified, (*e.g. recall* after 1000 images or *precision* after 20 images are retrieved). The values of *precision* and *recall* are often averaged, but it is important to know the basis on which this is done (Müller *et al.*, 1999).
- The rank of the best match (Berman and Shapiro, 1999) measures whether the «most relevant» image is in either the first 50 or first 500 images retrieved. 50 represents the number of images returned on a typical CBIRS screen and 500 is an estimate of the maximum number of images a user might look at when browsing.
- Similarly, *the average rank of relevant images* (Gargi and Kasturi, 1999) can give a good indication of the performance of a system, although it clearly contains less information than a PR graph. Moreover, it is vulnerable to outliers, since just one relevant image with a very high rank can adversely affect it. A simpler and more robust measure is the *rank of the first relevant image*, which is directly inspired from TREC and proves to be relevant in CBIR also.

Subsequent measures are reported in the literature and described below. Their analysis shows that they are in fact ad hoc measures which resemble the above. Their interpretation may therefore be difficult and we recommend to use standard measures when available.

- The *error rate* used in (Hwang *et al.*, 1999) is a typical measure in object or face recognition. It is in fact a single *precision* value, so it is important to know where the value is measured (see above). The error rate is defined as follows.

$$error\,rate = \frac{\#\text{ of non - relevant images retrieved}}{\text{Total }\#\text{ of images retrieved}} \qquad (2)$$

- The *retrieval efficiency* is defined in (Müller and Rigoll, 1999) as in Equation (3). It should be noted that, if the number of images retrieved is lower than or equal to the number of relevant images, this value equals the *precision*, otherwise it is the *recall* of a query. The interpretation of this value is therefore difficult since, depending on the context, it equals different values.

$$retrieval\ efficiency = \begin{cases} \dfrac{\#\text{ of relevant images retrieved}}{\text{Total }\#\text{ of images retrieved}} & if\,\#\text{ retrieved} > \#\text{ relevant} \\ \\ \dfrac{\#\text{ of relevant images retrieved}}{\text{Total }\#\text{ of relevant images}} & \text{otherwise} \end{cases} \qquad (3)$$

- The *correct and incorrect detection* are used in (Ozer *et al.*, 1999) as measures in an object recognition context. The numbers of correct and incorrect classifications are counted. When divided by the number of retrieved images, these measures are equivalent to *error rate* and *precision.*

Finally, the target testing approach differs significantly from other performance measures. Users are given a target image and the number of images which the user needs to examine before finding the target image is measured. Starting with random images, the user marks images as either relevant or non-relevant. This technique is employed in the PicHunter system (Cox *et al.*,1996). A more elaborate version of target testing is used in

(Müller *et al.*, 1999a), where the notion of moving targets is used to evaluate the ability of the system to track changes in user preferences during a query session.

Graphical representations: The above measures provide numerical factors, which are useful for a direct comparison of CBIRS. Defining graphical representations of such measures may also allow for determining further the characteristics or behaviour of a given system with respect to some test conditions. We list below some of the most used such graphical representation from which assessors can extract objective information (see later in this chapter for practical examples).

- *Precision vs. recall* (PR) graphs are a standard evaluation method in TR and are increasingly used by the CBIR community (Squire *et al.*, 1999a). PR graphs contain a lot of information, and their long use means that they can easily be interpreted by many researchers. In this respect, variations of such graphs should be avoided (*e.g.* a *recall vs. precision* graph). It is also common to present a *partial PR graph* (*e.g.* (He, 1997)). This can be useful in showing a region in more detail, but it can also be misleading since areas of poor performance can be omitted. Interpretation is also harder, since the scaling has to be watched carefully. A partial graph should therefore always be used in conjunction with the complete graph.
  One shortcoming of PR graphs is that they depend on the number of relevant images for a given query. Practical information such as *precision* or *recall* after a given number of images have been retrieved can also not be obtained from such a representation.
- *Precision vs. number of images retrieved* and *recall vs. number of images retrieved* graphs. Taken separately, these graphs contain only some of the information in a PR graph. When combined, however, they contain more information and can easily be interpreted. The *recall* graph looks more positive than a PR graph, especially when a few relevant images are retrieved late (Ratan *et al.*, 1999). The *precision* graph is similar to a PR graph, but it gives a better indication of what might be a good number of images to retrieve. It is more sensitive, however to the number of relevant images for a given query. If only part of the graph is shown it is hard to judge the performance (Aksoy and Haralick, 1999).
- *Correctly retrieved vs. all retrieved* graphs (Vascncelos and Lippman, 1999) contain the same information as *recall graphs*, but differently scaled. *Fraction correct vs. Number of images retrieved* graphs (Belongie *et al.*, 1998) are equivalent to *precision* graphs. *Average recognition rate vs. Number of images retrieved* graphs (Comaniciu, 1999) show the average percentage of relevant images among the first *N* retrievals, which is equivalent to the *recall* graph.
- *Retrieval accuracy vs. Noise* graphs are used in (Huet and Hancock, 1999) to show the change in retrieval accuracy as noise is added. A noisy image is used as a query and the rank of the original image is observed. This model does not correspond well to general CBIR applications but may be useful to complete the tests in some cases.

Proposals

In the preceding sections, we have investigated the various aspects of an evaluation context. While doing so, we already recommended the use of different standard components for ensuring the objectivity and usefulness of the evaluation.

First, we support the initiative of the University of Washington in Seattle for the construction of a common image database (Annotated DB, 1999). In terms of the development environment, we encourage the use of a communication protocol, which will permit the independent development of the user interface and the image search engine, as discussed in the next section.

Different evaluation factors and techniques have been described. It appears clearly that many of them are equivalent or highlight the same information. It would therefore be beneficial to the CBIR community if only standardised names were used for performance measures. Since the scaling or the use of partial graphs impedes interpretation, these techniques should only be used for emphasis, in conjunction with a complete graph. We propose a set of performance measures similar to those used in TREC. In addition to the PR graph, this set contains a mixture of rank-based, single-valued and graphical measures.

- $Rank_1$ and $Rank_N$: rank at which first relevant image is retrieved, normalised average rank of relevant images (see below and Equation(4)).
- $P(20)$, $P(50)$ and $P(N_R)$: *precision* after 20, 50 and N_R images are retrieved
- $R_P(.5)$ and $R(100)$: *recall* at *precision* .5 and after 100 images are retrieved
- PR graph

A simple average rank of relevant images is difficult to interpret, since it depends on both the collection size N and the number of relevant images N_R for a given query. Consequently, we propose the normalised average rank, $Rank_N$ defined in Equation (4).

$$Rank_N = \frac{1}{NN_R}\left(\sum_{i=1}^{N_R} R_i - \frac{N_R(N_R-1)}{2}\right), \tag{4}$$

where R_i is the rank at which the i^{th} relevant image is retrieved. This measure is 0 for perfect performance, and approaches 1 as performance worsens.

Examples of all these measures, using the same queries will be given in the next section when evaluating our CBIRS, *Viper* (see Figures 3 and 4, and Table 1). These results highlight the difference of information conveyed by the different performance measures we recommend.

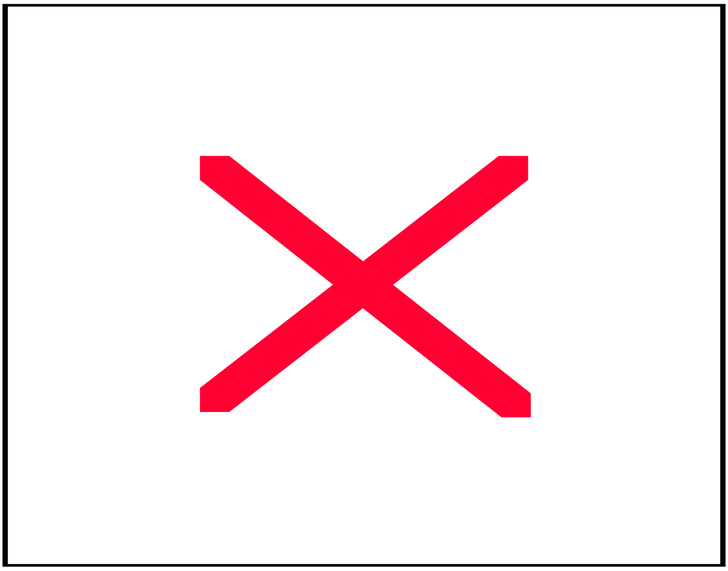
Viper : A CASE STUDY OF CBIRS

Viper is a CBIRS developed in the Computer Vision Group at the University of Geneva. It is based on the QBE search paradigm and has the following main characteristics.

- The retrieval system used is largely inspired from text retrieval systems in that it uses a very large number of simple features (as opposed to a small number of specialised image features).
- *Viper* uses several steps of positive and negative relevance feedback so as to allow the user to perform a directed search.
- Initially, *Viper* was designed with its own user interface. The collaboration with the CIRCUS group (http://lcavwww.epfl.ch/CIRCUS) at EPF Lausanne made apparent the need for sharing software components, in particular the interface. To this end, a new multimedia communication protocol called MRML (Multimedia Retrieval Markup Language) has been developed, in order to isolate *Viper* as a multimedia search engine, communicating with SnakeCharmer (figure 1), a JAVA-based interface shared by both the *Viper* and the CIRCUS groups.

The goal of *Viper* is twofold. It is primarily used as a platform for the development of an extensible generic CBIRS. In this respect, it has been designed so as to accept new features definitions and search algorithms easily. By this means, it should be easily

Figure 1: The Java interface SnakeCharmer connected to the Viper *CBIRS (http://viper.unige.ch)*



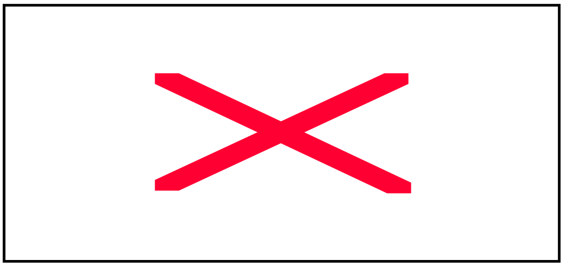
specialised in domains such as medical image search. Via the development of the *Viper* system, we also aim at developing an example platform for the benchmarking of CBIRS. MRML is now a vital part of our system and, conversely, our system is one example of the usage of MRML. The structure of *Viper* is sketched in Figure 2.

On top of these developments, we have also experimented in the direction of enhancing user interactivity via the study of *TrackingViper* (Müller *et al.*, 1999a), a CBIRS that accounts for user changes of mind during an image search. The base idea is to allow the user to perform a "directed visit" of the image search space (as opposed to perform a neighbourhood search). Finally, this platform also serves as test for the definition of heuristics for the feature selection, based on user interaction (Müller *et al.*, 2000a), (Müller *et al.*, 2000).

Image features

The *Viper* system, inspired by text retrieval systems, uses a very large number of simple features. The present version employs both local and global image colour and spatial frequency features, extracted at several scales, and their frequency statistics in both images

Figure 2: Structure of the Viper *system*



and the whole collection. The intention is to make available to the system low-level features which correspond (roughly) to those present in the human vision system.

More than 80,000 features are available to the system. Each image has $O(10^3)$ such features, the mapping from features to images being stored in an inverted file. The use of such a data structure, in conjunction with the feature weighting scheme discussed below, means that the integration of text annotations is completely natural: textual features can be treated in exactly the same way as visual ones.

Colour features: *Viper* uses a palette of 166 colours, derived by quantising *HSV* space into 18 hues, 3 saturations, 3 values and 4 grey levels. Two sets of features are extracted from the quantised image. The first is a colour histogram, where empty bins are discarded. The second represents colour layout. Each block in the image (the first being the image itself) is recursively divided into four equal-sized blocks, at four scales. The occurrence of a block with a given mode colour is treated as a binary feature. There are thus 56,440 possible colour block features, of which each image has 340.

Textural features: Gabor filters have been applied to texture classification and segmentation, as well as more general vision tasks (Ma and Manjunath, 1996), (Jain and Healey, 1998). We employ a bank of real, circularly symmetric Gabor filters, defined by

$$f_{mn}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma_m^2}} \cos\left(2\pi\left(u_{0m}x\cos\theta_n + u_{0m}y\sin\theta_n\right)\right), \qquad (5)$$

where m indexes filter scales, n their orientations, and u_{0m} gives the centre frequency. The half peak radial bandwidth is chosen to be one octave, which determines σ_m. The highest centre frequency is chosen as $u_{01} = 0.5$, and $u_{0m+1} = u_{0m}/2$. Three scales are used. The four orientations are: $\theta_0 = 0$, $\theta_{n+1} = \theta_n + \pi/4$. The resultant bank of 12 filters gives good coverage of the frequency domain, and little overlap between filters. The mean energy of each filter is computed for each of the smallest blocks in the image. This is quantised into 10 bands. A feature is stored for each filter with energy greater than the lowest band. Of the 27,648 such possible features, each image has at most 3,072. Histograms of the mean filter outputs are used to represent global texture characteristics.

Relevance feedback for enhanced user interaction

As discussed earlier, relevance feedback can produce a query which better represents a user's information need. *Viper* uses relevance feedback, in conjunction with feature weighting schemes inspired by those used in TR. Some modifications were necessary since the image features used cannot always be treated in the same way as words in documents.

The weighting used is based on the calculations of the *term frequency* tf_j (frequency of feature j in the image) and the *collection frequency* cf_j (frequency of the feature j in the entire database) of the feature, as well as its type (block or histogram). The motivation for using term frequency and collection frequency is very simple. Features with high *tf* characterise an image well; features with high *cf* do not distinguish that image well from others (Salton and Buckley, 1988). When considering a query q containing N images i with relevance indices R_i in {-1,0, 1}, the term frequency of feature j in the *pseudo*-image corresponding to q is then the weighted average of the image term frequencies.

$$tf_{qj} = \frac{1}{N}\sum_{i=1}^{N} tf_{ij} \cdot R_i \qquad (6)$$

From a query *q*, the aim is to calculate a score s_{qk} for each image *k* in the database. In *Viper*, this calculation makes use of a base weight bwf_{kqj}, defined for each feature *j* in the image *k*, with respect to the query *q*.

$$bwf_{kqj} = \begin{cases} tf_{qj} & \text{for block features} \\ \operatorname{sgn}(tf_{qj}).\min(\operatorname{abs}(tf_{qj}), tf_{kj}) & \text{for histogram features} \end{cases} \tag{7}$$

Note that the second case is a generalised histogram intersection.

For each feature *j*, a different logarithm factor is used, which depends upon the collection frequency cf_j of feature *j*.

$$lcf_j = \begin{cases} \log\left(\frac{1}{cf_j}\right) & \text{block} \\ 1 & \text{hist} \end{cases} \tag{8}$$

We investigated a variety of weighting functions in an earlier study (Squire *et al.*, 1999a). The weighting function used in the current version of the *Viper* system is defined as follows.

$$\text{classical inverse document frequency:} \quad wf_{kqj} = bwf_{kqj}.lcf_j^2 \tag{9}$$

Finally, for each image *k*, a score s_{qk} is calculated as

$$s_{kq} = \sum_j wf_{kqj} \tag{10}$$

The result of query *q* is then simply the list of images sorted with respect to the value of this score.

Communicating with *Viper*: MRML

Viper has the features of a CBIRS, based on the QBE paradigm. Following on the importance of being able to share software components, it has been our goal to separate the CBIR search engine and the interface itself. This research has lead to the development of the Multimedia Retrieval Markup Language (MRML). MRML is formally specified in (Müller *et al.*, 2000b). It provides a framework which separates query formulation from actual query shipping. It is designed to markup multi-paradigm queries for multimedia databases. MRML enables the separation of interface and query engine and thus eases their independent development.

MRML can be embedded into an existing system with little effort. First, it is XML-based, meaning that standard parsers can be used to process the communication messages. Further, the code for an example MRML-compliant CBIRS is freely-available (MRML, 2000) and provides the basic implementation of both ends of an MRML-based communication toolkit. MRML is currently in a testing phase at several universities and further applications based on this protocol such as benchmark tests and meta-query engines are under development.

MRML is designed to allow extension by independent groups. By this means, it provides a research platform for extensions which later may become a part of common MRML.

Features of MRML: MRML-based communications have the structure of a remote procedure call: the client connects to the server, sends a request, and stays connected to the server until the server breaks the connection. The server shuts down the connection after sending the MRML message which answers the request. This connectionless protocol has the advantage of easing the implementation of the server. MRML, in its current specification (and implementation) state, supports the following features:

- request of a capability description from the server,
- selection of a data collection classified by query paradigm; it is possible to request collections which can be queried in a certain manner,
- selection and configuration of a query processor, also classified by query paradigm; MRML permits the configuration of meta-queries during run time,
- formulation of QBE queries,
- transmission of user interaction data.

Graceful degradation - independent development on a common base: Graceful degradation is the key to successful independent extensions of MRML. The basic principles can be summarised as follows:

- servers and clients which do not recognise an XML element or attribute encountered in an MRML text should completely ignore its content,
- extensions should be designed so that all the standard information remains available to the generic MRML user.

These principles provide guidelines for independent extensions of MRML. To avoid conflicts between differing extensions of MRML, it is planned to maintain and promote a central database for the registration and documentation of MRML extensions. This would also facilitate the analysis of user logs which contain extended MRML.

MRML in practice: The following practical examples should give an overview of MRML in its current state of implementation. The sequence in which these examples are presented also give an idea of how a session between an MRML-compliant interface and an MRML-compliant search engine is organised.

Logging onto a CBIR server: An MRML server listens on a port for MRML messages on a given TCP socket. When connecting, the client requests the basic properties of the server, and waits for an answer.

```
<mrml> <get-server-properties /> </mrml>
```

The server then informs the client of its capabilities. This message is empty in the current version of MRML, but it allows for the extension of the protocol:

```
<mrml> <server-properties /> </mrml>
```

Using similar simple messages, the client can request a list of the collections available on the server, together with descriptions of the ways in which they can be queried. The client can then open a session on the server, and configure it according to the needs of its user (interactive client) or its own needs (*e.g.* meta-query agents). The client can also request the algorithms which can be used with a given collection:

```
<mrml> <get-algorithms  collection-id=”collection-1" /> </mrml>
```

This request is answered by sending the corresponding list of algorithms. This handshaking mechanism allows both interactive clients and programs (such as meta-query agents or automatic benchmarks) to obtain information describing the server.

In a similar simple manner, the client can open and close sessions for a user, and configure the algorithms chosen by the user. This enables multi-user servers and also on-the-fly learning by the query processor.

Interface configuration: The client can then request property sheet descriptions from the server. Different algorithms will have different relevant parameters which should be user-configurable (*e.g.* feature sets, speed vs. quality). *Viper*, for example, offers several weighting functions (Salton and Buckley, 1987) and a variety of methods for, and levels of, pruning (Squire *et al.*, 1999). All these parameters are irrelevant for CIRCUS. Thanks to MRML property sheets, the interface can adapt itself to these specific parameters. At the same time, MRML specifies the way the interface will turn these data into XML to send them back to the server.

Here is short example of interface configuration:

```
<property-sheet  property-sheet-id=”s1"
  type=”numeric” numeric-from=”1" numeric-to=”100"  numeric-step=”1"
  caption=”% features evaluated”
  send-type=”attribute”  send-name=”cui-percentage-features” />
```

This specifies a display element which will allow the user to enter an attribute with the caption «% of features evaluated». The values the user will be able to enter are integers between 1 and 100 inclusive. The value will be sent as an attribute cui-percentage-features=”33". This mechanism allows the use of complex property sheets, which can send XML text containing multiple elements. The interested reader is referred to (Müller *et al.*, 2000b) for details.

Query formulation: The query step is dependent on the query paradigms offered by the interface and the search engine. MRML currently supports QBE and has been designed to be extensible to other paradigms. A basic QBE query consists of a list of images and the corresponding relevance levels assigned to them by the user. In the following example, the user has marked two images, the image 1.jpg positive (user-relevance=”1") and the image 2.jpg negative (user-relevance=”-1"). All query images are referred to by their URLs.

```
<mrml session-id=”1" transaction-id=”44">
<query-step session-id=”1"  resultsize=”30"  algorithm-id=”algorithm-default”>
 <user-relevance-list>
 <user-relevance-element
   image-location=”http://viper.unige.ch/1.jpg” user-relevance=”1"/>
 <user-relevance-element
   image-location=”http://viper.unige.ch/2.jpg” user-relevance=”-1"/>
 </user-relevance-list>
</query-step>
</mrml>
```

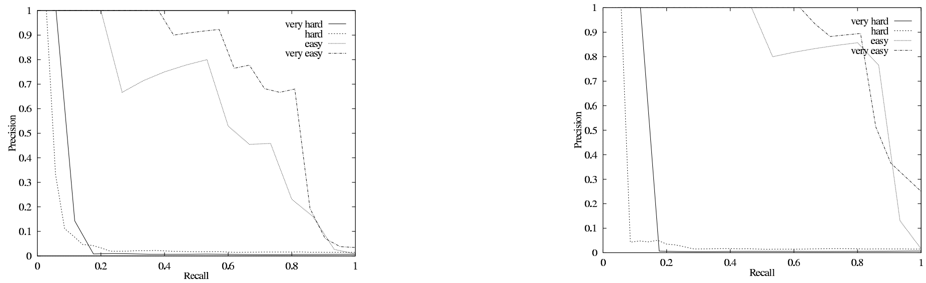
The server returns the retrieval result as a list of images, represented by their URLs. Queries can be grouped into transactions. This allows the formulation and logging of complex queries. This may be applied in systems which process a single query using a variety algorithms, such as the split-screen version of *TrackingViper* (Müller *et al.*, 1999a) or the system described by (Lee *et al.*, 1999). It is important in these cases to preserve in the logs the knowledge that two queries are logically related one to another.

Performance evaluation of *Viper*

We now use our CBIRS *Viper* to highlight and discuss an example use of the performance evaluation context defined throughout this study. In this section, we mostly aim at showing how performance measures and their graphical representation can be applied and interpreted. It is not our aim here to compare our system with any other (see next section) but rather to provide the reader with a base example of CBIRS evaluation.

We test our system using four types of queries, namely *very easy*, *easy*, *hard*, *very hard*, based on how well the query represents the expected results. Queries are made over a database of 2,500 images. Figure 3 first demonstrates that PR graphs can distinguish well

Figure 3: PR graphs for four different queries without and with feedback



(A)Without feedback

(B) With feedback

between differing results. These graphs also highlight the improvement made when, for example, relevance feedback is embedded into the querying paradigm.

One drawback is that the PR graph depends on the number of relevant images for a given query. We can see in both graphs that the plot for the *very hard* query starts later than the one for the *hard* query and looks better, although the decrease of the curve is much faster. Moreover, practical information such as *precision* or *recall* after a given number of images have been retrieved cannot be obtained from these graphs.

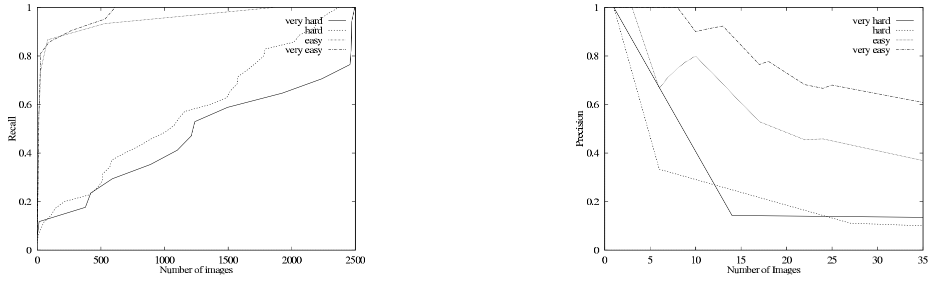
Therefore, graphs plotting the *precision* and *recall* versus the number of images are shown in figure 4. We can see in that the *recall graph* (figure 4 (A)) can distinguish well between the hard and easy queries, but not too well between the easy and very easy one. A complete *precision graph* does not contain much information in this case, and that is the reason for printing a partial one (figure 4 (B)). Here, we have the problem with the different numbers of relevant images like in the *PR graph*.

As a complement, the single-value measures detailed earlier in this study are displayed in Table 1.

Using this set of values, one can now distinguish between the type of queries and therefore better read the above graphs.

The above examples demonstrate the complementarity of the performance measures thus defined. Either of these gauges alone cannot lead to an objective evaluation. Similarly, the consistent use of a standard type of graph such as the PR graph allows for assessors to get used to reading such graphical representations of the evaluation and to extract further

Figure 4: Recall and precision versus number of images



(A) recall graph

(B) partial precision graph

Table 1: Performance measures for four different queries, over 2,500 images

Query	N_R	$Rank_1$	$Rank_N$	$P(20)$	$P(50)$	$P(N_R)$	$R_P(.5)$	$R(100)$
Very easy	21	1	0.028	0.73	0.51	0.71	0.82	0.86
Easy	15	1	0.067	0.47	0.23	0.60	0.62	0.87
Hard	35	5	0.426	0.20	0.08	0.10	0.05	0.14
Very hard	17	13	0.558	0.13	0.12	0.14	0.09	0.13

useful information. These examples pinpoint the need for a standard framework for the evaluation of CBIRS.

Comparing *Viper* with other CBIRS

As discussed earlier, an objective quantitative comparison of CBIRSs can only be made via the sharing of resources (mostly image databases and performance measures). This is clearly not the case at present and it is in fact one of the main goals of the *Viper* project to define a coherent framework in which CBIRS will be evaluated objectively.

As of now, it is only possible to compare CBIRS qualitatively, based on the characteristics they offer. To this end, reference (Viper, 2000) gives an updated list of systems that are publicly accessible. This list is clearly not exhaustive but may allow the reader to get an overview of the current state of research in CBIR.

CONCLUSION

In this study, we have investigated the design of a CBIRS. This made apparent the need for such tools, largely due to the wide availability of multimedia information on the WWW. The efficiency of search paradigms is however less well-defined and, in this work, we analysed the use of the QBE paradigm as a base for image retrieval. It was shown that this paradigm is a useful complement to other more widely used query paradigms, such as that based on keywords.

Further, we showed that the conception of a CBIRS calls for an expertise in a variety of domains, not only including computer vision but also software engineering, database management and human computer interaction. We then defined a context in which CBIRS may be evaluated in a consistent and objective fashion. The definition of this evaluation framework was based on our expertise acquired when developing the *Viper* system. One of the features we recommend for ensuring objective evaluations is the sharing of components, from the data to the software components. In this respect, we have developed and introduced MRML, which comes as a solution for guaranteeing the independence of CBIR software components. Further uses of this protocol include the development of a benchmark suite including a set of standard queries on a common database and the creation of a meta-search engine.

We also listed a set of performance measures that we derived from the experience acquired in the TR community and tested on our system. Results were given that demonstrated the usefulness of having different but complementary performance measures.

The analysis of current CBIRSs showed that there is still a real need for developing techniques that capture the semantic content of an image rather than just perform an analysis. Although progress have been made, computer vision is still a wide open issue after more than 20 years of research. Development of CBIRS should therefore be made in close relationship with advances made in the comprehension of the Human Visual System. Knowledge discovery and data mining techniques should still be used to fill as much as possible this gap between human notion of similarity and that defined by computer vision techniques. This can however only be performed via a constant and rigorous evaluation of the CBIRSs to highlight their strengths and weaknesses. This way, the interaction between the user and the multimedia databases will become more and more intuitive, thus increasing the usability and usage of CBIRS.

ACKNOWLEDGEMENTS

This project is supported by the Swiss National Foundation for Scientific Research under grant number 2000-052426.97.

REFERENCES

(Aksoy and Haralick, 1999) S. Aksoy and R. M. Haralick. Graph theoretic clustering for image grouping and retrieval. In (CVPR, 1999), pages 63–68.

(AltaVista, 2000) Altavista search engine. http://www.altavista.com, 2000.

(Annotated DB, 1999) Annotated groundtruth database. http://www.cs.washington.edu/research/imagedatabase/groundtruth/, 1999.

(Belongie *et al.*, 1998) S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the International Conference on Computer Vision (ICCV'98)*, Bombay, India, January 1998.

(Berman and Shapiro, 1999) A. P. Berman and L. G. Shapiro. Efficient content-based retrieval: Experimental results. In (CBAIVL, 1999), pages 55–61.

(Borgman, 1989) C. L. Borgman. All users of information retrieval systems are not created equal: an exploration into individual differences. *Information Processing and Management*, 25:225–250, June 1989.

(CBAIVL, 1999) *IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)*, Fort Collins, Colorado, USA, June 22 1999.

(Cleverdon *et al.*, 1966) C. W. Cleverdon, L. Mills and M. Keen. Factors determining the performance of indexing systems. Technical report, Cranfield Project, Cranfield, 1966.

(Comaniciu, 1999) D. Comaniciu, P. Meer, K. Xu, and D. Tyler. Retrieval performance improvement through low rank corrections. In (CBAIVL, 1999.

(COREL, 1999) Corel clipart & photos.
http://www.corel.com/products/clipartandphotos/, 1999.

(Cox *et al.*,1996) I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Target testing and the PicHunter Bayesian multimedia retrieval system. In *Advances in Digital Libraries (ADL'96)*, pages 66–75, Washington, D. C., 1996.

(Dy *et al.*, 1999) J. G. Dy, C. E. Brodley, A. Kak, C.-R. Shyu, and L. S. Broderick. The customized-queries approach to CBIR using using EM. In (CVPR, 1999), pages 400–406.

(Gargi and Kasturi, 1999) U. Gargi and R. Kasturi. Image database querying using a multiscale localized color representation. (CBAIVL, 1999).

(Gevers and Smeulders, 1996) T. Gevers and A. W. M. Smeulders. A comparative study of several color models for color image invariants retrieval. In *Proceedings of the First International Workshop ID-MMS'96*, pages 17–26, Amsterdam, The netherlands, August 1996.

(Güting, 1994) R. H. Güting. An introduction to spatial database systems. *VLDB Journal*, 3(4):1–2, 1994.

(He, 1997) O. He. An evaluation on MARS - an image indexing and retrieval system. Research report, University of Illinois, Urbana-Champaign, USA, 1997.

(Huet and Hancock, 1999) B. Huet and E. R. Hancock. Inexact graph retrieval. In (CBAIVL, 1999), pages 40–44.

(VISUAL, 1999) D. P. Huijsmans and A. W. M. Smeulders, editors. *Third International Conference On Visual Information Systems (VISUAL'99)*, in Lecture Notes in Computer Science 1614, Amsterdam, The Netherlands, 1999.

(Hwang *et al.*, 1999) W.-S. Hwang, J. J. Weng, M. Fang, and J. Qian. A fast image retrieval algorithm with automatically extracted discriminant features. In (CBAIVL, 1999), pages 8–12.

(CVPR, 1999) IEEE Computer Society. *Proceedings of the 1999 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, Colorado, USA, June 23–25 1999.

(Jain and Healey, 1998) A. Jain and G. Healey. A multiscale representation including opponent color features for texture recognition. *IEEE Transactions on Image Processing*, 7(1):124–128, January 1998.

(Jain, 1989) A. K. Jain. *Fundamentals of digital image processing*. Prentice-Hall International, London, 1989.

(Jain and Vailaya, 1996) A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, August 1996.

(Kass *et al.*, 1987) M. Kass, A. Witkin and D. Terzopoulos. Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

(Lee *et al.*, 1999) C. S. Lee, W.-Y. Ma and H.J. Zhang. Information Embedding Based on User's Relevance Feedback for Image Retrieval. In Panchanathan *et al.* (SPIE, 1998).

(Lindeberg, 1994) T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

(Ma and Manjunath, 1996) W. Y. Ma and B. S. Manjunath. Texture features and learning similarity. In *Proceedings of the 1996 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 425–430, San Francisco, California, June 1996.

(Markkula and Sormunen, 1998) M. Markkula and E. Sormunen. Searching for photos - journalists' practices in pictorial IR. In J. P. Eakins, D. J. Harper and J. Jose, editors, *The Challenge of Image Retrieval, A Workshop and Symposium on Image Retrieval*, Electronic Workshops in Computing, Newcastle upon Tyne, 5–6 February 1998. The British Computer Society.

(MPEG, 1998) MPEG Requirements Group. MPEG-7: Context and objectives (version 10 Atlantic City). Doc. ISO/IEC JTC1/SC29/WG11, International Organisation for Standardisation, October 1998.

(MRML, 2000) Multimedia Retrieval Markup Language. http://www.mrml.net.

(Müller *et al.*, 2000) H Müller, W Müller, S Marchand-Maillet, D McG. Squire, and T. Pun. Learning features weights from user behaviour in Content-Based Image Retrieval. In *ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining*, Boston, August 2000.

(Müller *et al.*, 2000a) H. Müller, W. Müller, D. McG. Squire, S. Marchand-Maillet, and T. Pun. Strategies for positive and negative relevance feedback in image retrieval. In *International Conference on Pattern Recognition (ICPR'2000),* Barcelona, September 2000.

(Müller *et al.*, 2000b) W. Müller, H. Müller, S. Marchand-Maillet, T. Pun, D. McG Squire, Z. Pecenovic, C. Giess and A. P. de Vries. MRML: A Communication Protocol for Content-Based Image Retrieval (MRML, 2000) . *In International Conference on Visual Information System(VISUAL'2000)*, Lyon, France, November 2000.

(Müller *et al.*, 1999) H. Müller, W. Müller, D. McG. Squire, and T. Pun. Performance evaluation in content-based image retrieval: Overview and proposals. Technical Report 99.05, Computer Vision Group, CUI, University of Geneva, Switzerland, 1999.

(Müller and Rigoll, 1999) S Müller and G. Rigoll. Improved stochastic modeling of shapes for content-based image retrieval. In (CBAIVL, 1999).

(Müller *et al.*, 1999a) W. Müller, D. McG. Squire, H. Müller, and T. Pun. Hunting moving targets: an extension to Bayesian methods in multimedia databases. In Panchanathan *et al.* (SPIE, 1998). (SPIE Symposium on Voice, Video and Data Communications).

(Nielsen, 1993) Jakob Nielsen, *Usability Engineering*. Academic Press, Boston, MA, 1993.

(Over, 1998) Paul Over. A review of Interactive TREC. In *MIRA* workshop, Dublin, Ireland, October 1998.

(Ozer *et al.*, 1999) Burak Ozer, Wayne Wolf and Ali N. Akansu. A graph based object description for information retrieval in digital image and video libraries. In (CBAIVL, 1999), pages 79–83.

(SPIE, 1998) Sethuraman Panchanathan, Shih-Fu Chang and C.-C. Jay Kuo, editors. *Multimedia Storage and Archiving Systems IV (VV02)*, volume 3846 of *SPIE Proceedings*, Boston, Massachusetts, USA, September 20–22 1999.

(Pun and Squire, 1996) Thierry Pun and David McG. Squire. Statistical structuring of pictorial databases for content-based image retrieval systems. *Pattern Recognition Letters*, 17:1299–1310, 1996.

(QBIC, 1998) QBIC^{TM} – IBM's Query By Image Content. http://wwwqbic.almaden.ibm.com/qbic/, 1998.

(Rabiner and Huang, 1993) L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliffs, NJ, 1993.

(Ratan *et al.*, 1999) A. L. Ratan, O. Maron, W. E. L. Grimson, and T. Lozano-Perez. A framework for learning query concepts in image classification. In (CVPR, 1999), pages 423–429.

(Rowley *et al.*, 1998) H. A. Rowley, S. Baluja and T. Kanade. Neural-Network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-20(1):23–38, 1998.

(Salton, 1971) G. Salton. Evaluation parameters. (Salton, 1971a), page 55–112.

(Salton, 1971a) G. Salton. *The SMART* Retrieval System, Experiments in Automatic Document Processing. Prentice Hall, Englewood Cliffs, USA, 1971.

(Salton, 1992) G. Salton. The state of retrieval system evaluation. *Information Processing and Management*, 28(4):441–450, 1992.

(Salton and Buckley, 1987) G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical Report 87-881, Department of Computer Science, Cornell University, Ithaca, New York 14853-7501, 1987.

(Salton and Buckley, 1988) G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

(Schmid and Mohr, 1997) C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

(Sclaroff *et al.*, 1997) S. Sclaroff, L. Taycher and M. La Cascia. ImageRover: a content-based browser for the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, San Juan, PuertoRico, 1997.

(Sethian, 1999) J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

(Shyu *et al.*, 1999) C.-R. Shyu, A. Kak, C. Brodley, and L. S. Broderick. Testing for human perceptual categories in a physician-in-the-loop CBIR system for medical imagery. In (CBAIVL, 1999), pages 102–108.

(Spark Jones and Rijsbergen, 1975) K. Sparck Jones and C. J. van Rijsbergen. Report on the need for and provision of an ideal information retrieval test collection. British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

(Squire *et al.*, 1999) D McG. Squire, H Müller and W Müller. Improving response time by search pruning in a content-based image retrieval system, using inverted file techniques. In (CBAIVL, 1999), pages 45–49.

(Squire *et al.*, 1999a) D. McG. Squire, W. Müller and H. Müller. Relevance feedback and term weighting schemes for content-based image retrieval. In Huijsmans and Smeulders (VISUAL, 1999), pages 549–556.

(Squire *et al.*, 1999b) D. McG. Squire, W. Müller, H. Müller, and J. Raki. Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In *The 11th Scandinavian Conference on Image Analysis (SCIA'99)*, pages 143–149, Kangerlussuaq, Greenland, June 7–11 1999.

(Squire *et al.*, 1997) D. McG. Squire and T. Pun. A comparison of human and machine assessments of image similarity for the organization of image databases. In M. Frydrych, J. Parkkinen and A. Visa, editors, *The 10th Scandinavian Conference on Image Analysis (SCIA'97)*, pages 51–58, Lappeenranta, Finland, June 1997.

(TREC, 1999) Text REtrieval Conference (TREC). http://trec.nist.gov/, 1999.

(Tversky, 1977) A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, July 1977.

(Rijsbergen, 1979) C. J. van Rijsbergen. Evaluation. In *Information Retrieval*, chapter 7, pages 112–123. Prentice Hall, Englewood Cliffs, NJ, USA, 1979.

(Vascncelos and Lippman, 1999) Nuno Vasconcelos and Andrew Lippman. Probabilistic retrieval: new insights and experimental results. In (CBAIVL, 1999), pages 62–66.

(Vellaikal and Kuo, 1998) A. Vellaikal and C.-C. J. Kuo. Hierarchical clustering techniques for image database organization and summarization. In (SPIE, 1998).

(Vendrig *et al.*, 1999) J. Vendrig, M. Worring and A. W. M. Smeulders. Filter image browsing: Exploiting interaction in image retrieval. In Huijsmans and Smeulders (VISUAL, 1999), pages 147–154.

(Viper, 2000) Links to CBIR systems, http://viper.unige.ch/other_systems.html

(VisTex, 1995) VisTex: Vision texture database. http://whitechapel.media.mit.edu/vismod/ , 1995.

(Vorhees and Harmann, 1998) E. M. Vorhees and D. Harmann. Overview of the seventh text retrieval conference (TREC-7). In *The Seventh Text Retrieval Conference*, pages 1–

23, Gaithersburg, MD, USA, 1998.

(White and Jain, 1996) D. A. White and R. Jain. Algorithms and strategies for similarity retrieval. Technical Report VCL-96-101, Visual Computing Laboratory, University of California, July 1996.

(Wieckert, 1998) J. Wieckert. *Anisotropic Diffusion in Image Processing*. European Consortium for Mathematics in Industry, Stuttgart, Germany, 1998. (Winter and Nastar, 1999) A. Winter and C. Nastar. Differential feature distribution maps for image segmentation and region queries in image databases. In (CBAIVL, 1999), pages 9–17.

(Witten *et al.*, 1994) I. H. Witten, Al. Moffat and T. C. Bell. *Managing gigabytes: compressing and indexing documents and images*. Van Nostrand Reinhold, 115 Fifth Avenue, New York, NY 10003, USA, 1994.

(Zobel and Moffat, 1998) J. Zobel and A. Moffat. Exploring the similarity space. *ACM SIGIR Forum (SIG on Information Retrieval)*, 32(1):18–34, 1998.