Journal of Visual Languages and Computing (2001) **12,** 81–103
doi:10.1006/jvlc.2000.0188, available online at http://www.idealibrary.com on IDEAL®

Quicklook^2: An Integrated Multimedia System

G. Ciocca, I. Gagliardi and R. Schettini*

Istituto Tecnologie Informatiche Multimediali, Consiglio Nazionale delle Ricerche, Via Ampere 56, 20131 Milano, Italy. E-mail: (ciocca,isabella,centaura)@itim.mi.cnr.it

Received 11 April 2000; accepted 30 August 2000

The need to retrieve visual information from large image collections is shared by many application domains. This paper describes the main features of the multimedia information retrieval engine of Quicklook^2. Quicklook^2 allows the user to query image and multimedia databases with the aid of sample images, or an impromptu sketch and/or textual descriptions, and progressively refine the system's response by indicating the relevance, or non-relevance of the retrieved items. The major innovation of the system is its relevance feedback mechanism that performs a statistical analysis of both the image and textual feature distributions of the retrieved items the user has judged relevant, or not relevant to identify what features the user has taken into account (and to what extent) in formulating this judgement, and then weigh their influence in the overall evaluation of similarity, as well as in the formulation of a new, single query that better expresses the user's multimedia information needs. Another important contribution is the design and integration with the relevance feedback mechanism of an indexing scheme based on triangle inequality to improve retrieval efficiency. The performance of the system is illustrated with examples from various application domains and for different types of queries (target search as well as similarity search).

1. Introduction

The retrieval of visual information from large image collections is a critical issue in many application domains [1, 2]. This paper presents the main features of Quicklook^2, a general-purpose system that combines in a single framework three approaches usually considered alternative for querying image databases: the alphanumeric relational query, the content-based image query utlizing automatically computed low-level image features (such as color and texture), and the textual similarity query exploiting any textual annotations attached to database images (such as figure captions or textual cards, etc.).

These approaches are complementary: all of them can be useful in a query session to deal with the different types of information that a general-purpose visual information retrieval system should be able to deal with [3]:

- *Content-independent data*: Data not directly concerned with image content, but are in some way related to it. Examples are the format, the author, date, location, ownership. This type of data is handled with traditional DBMS.

*Corresponding author

1045-926X/01/020081+23 $35.00/0

- *Content-dependent data*: The low-intermediate-level features that refer to the visual content of images. Examples are color, texture, shape, spatial relationship, and combinations of these. This type of information is very difficult to define and index using natural language: it requires the design of algorithms that extract suitable surrogates of the visual features from the images; these surrogates can then be processed automatically.
- *Content-descriptive data*: Data concerning image semantics. This type of information is handled with information retrieval tools.

Quicklook^2 provides a single framework for managing all these different types of information in an integrated way, and copes with various retrieval tasks, including:

1. *Target search*: The user knows exactly what image he is looking for.
2. *Similarity search*: The user wants to retrieve all the images that resemble an example. As similarity is a rather fuzzy term, we do not make any distinction here between perceptual similarity (the query and the retrieved items are similar in appearance, but may depict different subjects) and semantic similarity, in which the query and the retrieved items may appear perceptually different, but belong to the same category as semantically defined.

The evaluation of multimedia information is subjective in general, and that of visual and semantic information in particular. We can hope to have some chance of success in processing the different types of queries defined above, with a general-purpose system only if all available information, both visual and textual, is used in indexing, and if user feedback is considered in the retrieval process in order to understand what features the user has taken into account (and to what extent) in judging an item relevant or not relevant. We have designed Quicklook^2 to address these issues. We have also taken into account the fact that users find it much easier to provide examples that match, or do not match his/her information needs, than to explicitly describe them. With Quicklook^2, the database can be queried with the aid of sample images, or user-made sketches, and/or textual descriptions. When a query is submitted to the system, the retrieved items are presented in decreasing order of relevance, and the user is then allowed to progressively refine the system's response by indicating their relevance, or non-relevance.

Obviously this requires a suitable data structure to ensure system efficiency. Quicklook^2 implements an indexing scheme based on triangle inequality [4], which reduces the number of direct comparisons between the images, i.e. feature vectors representing the query and those representing the database items. The integration of this indexing scheme with the relevance feedback mechanism is another contribution of our research.

Our presentation of Quicklook^2 is organized as follows: Section 2 contains an overview of related studies and existing systems. In Section 3, we describe how the visual and textual features are extracted and used in image indexing. Section 4 describes the relevance feedback algorithm implemented. In Section 5, we outline the overall system architecture, and report on some experimental results.

2. Related Works

In the framework of a visual information retrieval system, the integrated use of content-independent data is a rather straightforward affair: queries are generally made in

standard query languages, such as SQL, to reduce the number of database items that must be further evaluated. General-purpose systems, such as QBIC [5], VIR [6], and NETRA [7] regularly employ this feature, as does our system.

Much more challenging is the effective and efficient use of content-dependent data automatically extracted from the images themselves. Many special issues of leading journals have been dedicated to this topic, and several surveys have been published in recent years [1, 8, 9]. Notwithstanding the substantial progress made in this direction, this approach appears truly feasible only for retrieving images from thematic databases, where the semantic content is limited to a specific domain. Although several general-purpose systems have also been developed in the last few years, the integrated management of the various image features remains complex and application dependent [10, 11]. Several factors may intervene when choosing the aggregation operator to integrate the results of a query based on single features [12]: different tasks in the same context deal with similarity at different levels of precision; similarity depends greatly on the nature of the objects to which it is applied, and on the features selected for their description; different users from different backgrounds may interpret image content differently, and the objective of their queries may also differ. All these factors, which are interrelated and consequently influence each other, make it quite impossible to determine in advance the most suitable aggregation operator for the different similarity measures, e.g. [13]. This leaves to the users the burden of formulating their information needs, which may be rather difficult (and tiresome) to express as a weighted combination of the features that are actually used for retrieval [9].

In some systems the capability of retrieving similar images by semantic contents is achieved by exploiting textual annotations, which are manually associated with the images. More challenging is the association of significant terms, or keywords to images in a completely unsupervised manner [14]. Still another approach has been used in [15], where keywords are automatically associated to a video, extracting them from closed-captions.

Much research has concerned the automatic assignment of significant terms to images in WWW pages on the basis of the different parts that can be identified through the use of HTML tags. Harmandas, for example, uses the sections after the image's URL to extract terms to associate with the image to index [16]. In the AMORE system the keywords associated with an image are extracted from different sources, such as the image URL, the ALT text, the heading, and the title [17]. La Cascia *et al.* [18] use latent semantic indexing (LSI) to identify the context in which an image appears: words appearing in the HTML document are extracted, and weights assigned to them according to the tag in which the words appear (e.g. headers, title, ALT, etc.). Ortega *et al.* [19] present WebMARS, an integrated textual and visual search engine for Web documents, for textual data they support two granularities, at the local level and the document level.

Once textual indices are in some manner associated with the images, their similarity function must be defined. The basic idea of finding pieces of text similar to a given one has been exploited in the framework of the hypertext for the automatic generation of the hypertextual link [20–23]. In 1995 a workshop on 'IR and the Automatic Construction of Hypermedia' was held during the ACM SIGIR conference, and in 1997 the authoritative journal *IP&M* published a monographic issue on the subject [24]. Several experiments have been dedicated to the matter, but few prototypes have been produced for an integrated multimedia environment.

It is obvious that user feedback must be considered in the retrieval of multimedia information. The potentials of relevance feedback in textual information retrieval have been widely studied. In image retrieval, it has been employed by Minka and Picard [25] and by Cox *et al.* [26] for target search, and by Rui *et al.* [27], La Cascia *et al.* [18] for similarity retrieval. In Ciocca and Schettini [28] we designed an algorithm that, through the statistical analysis of the image feature distributions of the retrieved images the user has judged relevant, or not relevant, identifies what features the user has taken into account in formulating this judgement. It then modifies the contribution of the different features in the overall evaluation of image similarity. This algorithm is further extended here to cope with different types of queries (target search as well as similarity search), and with the true integration of multiple visual and/or textual features in the query of the database.

Several image retrieval systems are now available, but none of these seems to have all the functionalities and flexibility of Quicklook^2. Most of the advanced research systems have very sophisticated image indexing and retrieval algorithms, but have none or very limited integration with textual indexing. Commercial image search engines such as QBIC [5], VIR [6] and VRW [8], designed as general purpose tools, and available as add-ons of existing database management systems such as Oracle or Informix, serve as application development tools, rather than stand-alone, general purpose, multimedia retrieval systems. One system feature found only in research systems and not in commercial ones, is relevance feedback. This is probably due to the fact that it is computationally quite expensive unless suitable data structures are available.

3. Image Indexing

3.1. Using Pictorial Features

Because perception is subjective, there is no one 'best' representation of image contents whatever the type of database to be indexed. Since we did not design Quicklook^2 for a particular application, we have constituted a general-purpose library of low-level features to use in image indexing. This library is continuously extended and updated with new features. The features implemented in the present version are:

1. The ratio between the dimensions of the images.
2. The color histogram and color coherence vectors (CCV) in the CIELAB color space quantized in 64 colors [29]; the CVV buckets color pixels as coherent or incoherent according to whether or not they belong to a large, similarly colored region. Before computing the CCV the image is blurred by local averaging in a 3×3 neighborhood.
3. The histogram of the transition in color (in a CIELAB color space quantized in 11 colors, namely, red, orange, yellow, green, blue, purple, pink, brown, black, gray and white) [30].
4. The spatial chromatic histogram (SCH), summarizing information about the location of pixels of similar color and their arrangement within the image [31].
5. The moments of inertia (mean, variance, skewness, and kurtosis) of the color distribution in the CIELAB space [32].

6. A histogram of opportunely filtered contour directions (considering only high gradient pixels); the edges are extracted by Canny edge detectors, and the corresponding edge directions quantized in 72 bins at 2.5° intervals. To compensate for differences in image size, the histograms are normalized with respect to the total number of edge pixels detected in the image [33].
7. The statistical information on image edges extracted by the Canny algorithm: (i) the percentage of low, medium, and high contrast edge pixels in the image; (ii) the parametric thresholds on the gradient strength corresponding to medium and high contrast edges; (iii) the number of connected regions presenting closed high contrast contours; and (iv) the percentage of medium contrast edge pixels connected to high contrast edges [34].
8. The mean and variance of the absolute values of the coefficients of the sub-images at the first three levels of the multi-resolution Daubechies wavelet transform of the luminance image [30].
9. The Hu invariant moments [35].
10. The spatial composition of the color regions identified by the process of quantization in 11 colors: (i) fragmentation (the number of color regions), (ii) distribution of the color regions with respect to the center of the image; (iii) distribution of the color regions with respect to the x axis, and with respect to the y-axis [30].
11. The estimation of statistical features based on the neighborhood gray-tone difference matrix (NGTDM), i.e. coarseness, contrast, busyness, complexity, and strength [36, 37].
12. The percentage of pixels that correspond to skin according to a detector trained on a large amount of labeled skin data, e.g. [38].

The SCH features are compared using the distance metric proposed in [31]. The city-block distance measure L_1 is used to compare all other features, as it is statistically more robust than the Euclidean distance measure L_2 [39]. The non-normalized distance for a generic feature F_h, having c components, is therefore computed as follows:

$$D(F'_h, F''_h) = \sum_{i=1}^{c} |F'_h(i) - F''_h(i)| \tag{1}$$

while given two spatial chromatic histograms H' and H'' having c bins, the distance is computed as follows:

$$D(H', H'') = \sum_{i=1}^{c} \min\left(h_{H'}(i) - h_{H''}(i)\right) \times \left(\frac{\sqrt{2} - d(\mathbf{b}_{H'}(i),\ (\mathbf{b}_{H''}(i))}{\sqrt{2}} + \frac{\min(\sigma_{H'}(i), \sigma_{H''}(i))}{\max(\sigma_{H'}(i), \sigma_{H''}(i))}\right) \tag{2}$$

where $h(i)$ is the ratio of pixels having color i, $\mathbf{b}$ are the relative coordinates of the baricenter of their spatial distribution, and σ is the corresponding standard deviation.

When a new database has to be created the user chooses the features to use in indexing the pictorial contents of the images. When searching the database the user can

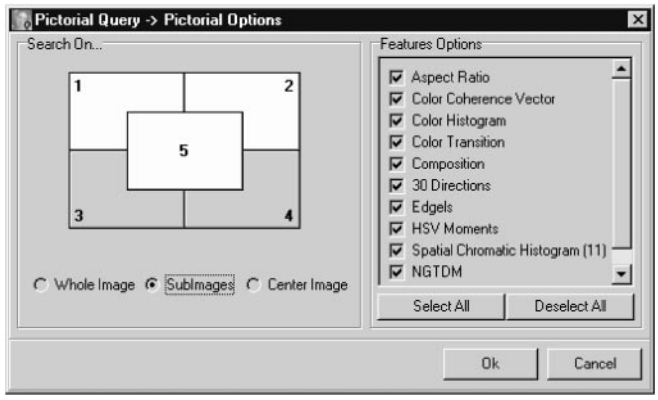


Figure 1. Window for feature selection

select the set of features to use in evaluating the similarity, and decide whether these are referred to the global image and/or to sub-images obtained by dividing the original image in different ways as shown in Figure 1.

3.2. Using Textual Annotations

Images are sometimes accompanied by textual annotations describing their semantic contents. These annotations can be used in indexing and retrieving images if significant terms are automatically extracted from them to create a dictionary, and a suitable similarity function among sets of significant terms is defined [40, 41].

Generally speaking, dictionaries contain the sets of significant terms that can be used to index textual annotations, and they can be created in two opposite ways: automatically, as the result of an IR process using lists of stop-words (lists of 'poor discriminators', that is terms, such as articles or adverb, that are too frequent to be significant), or manually, by experts in the domain who indicate the more significant terms according to the criteria applied.

In our system, designed to be general purpose, the dictionaries are created automatically, and are composed of all the terms present in the textual annotations (excepting those on a standard Italian stop list). No stemming procedure is applied, as no satisfactory algorithm is available for the Italian language. Most morphological variations (singular/plural, feminine/masculine, etc.) are, however, automatically eliminated.

Each index term of a document is automatically assigned a weight TW reflecting its importance, on the basis of the number of times the term occurs in the document, and in the entire archive. For example, the weight TW of the term k in document i is computed as follows:

$$TW_{ik} = Freq_{ik}\left(1 + \log\frac{n}{DocFreq_k}\right) \tag{3}$$

where $Freq_{ik}$ is the frequency of term i in document k, n is the total number of documents in the database, and $DocFreq_k$ is the number of documents in which term k occurs at least once.

The textual annotation associated with the generic image i is therefore indexed by a set of its relevant terms to which TW weights have been assigned. We call such a set T_i. Text similarity, TS, between the textual annotations T_i and T_j, is defined as follows:

$$TS(T_i, T_j) = \frac{\sum_{k \in (T_i \cap T_j)} (TW_{ik} TW_{jk})}{\sqrt{\sum_{k \in T_i} (TW_{ik})^2 \sum_{k \in T_j} (TW_{ik})^2}} \tag{4}$$

TS can assume any value in the range of [0, 1]. The greater the value of TS, the greater the similarity between the two textual annotations.

4. Combing Distance Measures with Relevance Feedback

The key concept of the relevance feedback mechanism, defined in [28], is that the statistical analysis of the feature distributions of the images the user has judged relevant, or not relevant, can be used to identify the features the user has taken into account (and to what extent) in formulating this judgement, and then accentuate the influence of these features in the overall evaluation of image similarity, as well as in the formulation of a new query. Its major virtue is that of being truly description-independent: the index can be modified, or extended to include other features, without requiring any change in the algorithm.

Sub-vectors of visual features, the color histogram for example, are indicated by $\mathbf{X}_h^i$, where i is the vector index, and h, the index of the feature; $\mathbf{T}^i$ is the corresponding textual annotation, if available. The contribution of different visual features to overall image similarity could be compared with different metrics (e.g. L1 or L2 distances, but also metrics developed ad hoc for a given feature). The relevance feedback mechanism takes this into account, and is truly independent of the distances used to evaluate single features. We will indicate here with D_h the distance associated with the feature hth; and with TS, the similarity function associated to the textual annotations, as defined in the previous section. The global metric used to evaluate the dissimilarity between two database items is defined as a linear combination of the distances between the individual features:

$$\text{Dissimilarity}\,(\mathbf{X}^i, \mathbf{X}^j) = \frac{1}{p} \sum_{h=1}^{p} w_h D_h(\mathbf{X}_h^i, \mathbf{X}_h^j) + w_T(1 - TS(\mathbf{T}^i, \mathbf{T}^j)) \tag{5}$$

in which p is the number of visual features considered, while w_h and w_T are weights.

There are two drawbacks to this formulation of image dissimilarity that had to be taken into account in designing Quicklook[2]:

1. The single distances may be defined on intervals of widely varying values: if we do not want one feature to overshadow the others simply because of its magnitude, we must normalize the distances to a common interval so that equal emphasis is placed on every feature score.
2. The weights must often be set heuristically by the user, and this may be rather difficult, as there may be no clear relationship between the features used to index

the image database and those evaluated by the user in a subjective image similarity evaluation.

4.1. Normalization of Distances Between Visual Features

To cope with the problem of distances defined on different intervals of values, we use the following normalization derived from the Gaussian normalization [42].

Assuming that the database contains n images, the average distance between the visual features of database items and the standard deviation are computed as follows:

$$\mu_h = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} D_h(\mathbf{X}_h^i, \mathbf{X}_h^j) \tag{6}$$

$$\sigma_h = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} [D_h(\mathbf{X}_h^i, \mathbf{X}_h^j) - \mu_h]^2 \tag{7}$$

The vector of the normalized distance between two images having indices i and j, respectively, is

$$\begin{aligned} \mathbf{D}(\mathbf{X}^i, \mathbf{X}^j) &= \left[\frac{D_1(\mathbf{X}_1^i, \mathbf{X}_1^j)}{\mu_1 + \mathrm{K}\sigma_1}, \ldots, \frac{D_h(\mathbf{X}_h^i, \mathbf{X}_h^j)}{\mu_h + \mathrm{K}\sigma_h}, \ldots, \frac{D_p(\mathbf{X}_p^i, \mathbf{X}_p^j)}{\mu_p + \mathrm{K}\sigma_p} \right]^{\mathrm{T}} \\ &= [d_1(\mathbf{X}_1^i, \mathbf{X}_1^j), \ldots, d_h(\mathbf{X}_h^i, \mathbf{X}_h^j), \ldots, d_p(\mathbf{X}_p^i, \mathbf{X}_p^j)]^{\mathrm{T}} \end{aligned} \tag{8}$$

where K is a positive constant that influences the number of out-of-range values. In our experiments K was set at 3. Any out-of-range values are mapped to the extreme values, so that they do not bias further processing. At this point our dissimilarity function has the following form:

$$\begin{aligned} \mathrm{Dist}(\mathbf{X}^i, \mathbf{X}^j) &= \frac{1}{p} \sum_{h=1}^{p} w_h d_h(\mathbf{X}_h^i, \mathbf{X}_h^j) + w_T(1 - TS(T_i, T_j)) \\ &= \frac{1}{p} \sum_{h=1}^{p} w_h D_h(\mathbf{X}_h^i, \mathbf{X}_h^j) + w_T d_T \end{aligned} \tag{9}$$

It can be argued that textual similarity should by default have a higher weight than pictorial similarity. We have adopted this somewhat conservative approach for the following reasons:

1. Text descriptions reflect the point of view of the annotator, which may not be that of the final user querying the system;
2. In a large database there is a high risk of inconsistency in textual annotations, as several different annotators may be engaged in this task; and
3. The relevance feedback allows rapid tuning of the similarity measure on the basis of just a few examples. This formulation of image similarity reduces the risk of over-learning the user's notion of similarity, and jamming the system.

4.2. Estimation of Weights

To estimate the weights we let $\mathbf{R}^+$ be the set of relevant items selected by the user ($\mathbf{R}^+$ is usually only an approximation of the set of items relevant to the query in the whole database), $\mathbf{d}_h^+$, the set of normalized distances (computed on the visual feature h) among the elements of $\mathbf{R}^+$, and μ_h^+, the mean of the values of $\mathbf{d}_h^+$. Similarly, we let $\mathbf{d}_T^+$ be the set of normalized distances (computed on the textual indices) among the elements of $\mathbf{R}^+$, and μ_T^+, the mean of the values of $\mathbf{d}_T^+$.

We define $\mathbf{R}^-$ as the set of non-relevant items selected by the user to serve as negative examples, while $\mathbf{d}_h^-$ and $\mathbf{d}_T^-$ are the corresponding sets of distances. From $\mathbf{R}^+$ and $\mathbf{R}^-$ we can then determine whether the influence of a feature must be limited by reducing the corresponding weight in computing the dissimilarity: we let $\mathbf{R}^\pm$ be the union of $\mathbf{R}^+$ with $\mathbf{R}^-$, and $\mathbf{d}_h^\pm$, $\mathbf{d}_T^\pm$ the corresponding sets of distances among its elements. Since we cannot make any assumptions about the statistical distribution of the features of non-relevant images by analyzing $\mathbf{R}^-$ (the selected non-relevant images may not be representative of all the non-relevant images in the database), we exclude set $\mathbf{d}_h^-$ from $\mathbf{d}_h^\pm$, and set $\mathbf{d}_T^-$ from $\mathbf{d}_T^\pm$ obtaining two new sets of distances: $\mathbf{d}_h^* = \mathbf{d}_h^\pm \backslash \mathbf{d}_h^-$ and $\mathbf{d}_T^* = \mathbf{d}_T^\pm \backslash \mathbf{d}_T^-$.

The weight terms w_h and w_T used in the following equations ($x = h$ or T) are then updated:

$$w_x^+ = \begin{cases} \dfrac{1}{\varepsilon} & \text{if } |\mathbf{R}^+| < 3 \\ \dfrac{1}{\varepsilon + \mu_x^+} & \text{otherwise} \end{cases} \tag{10}$$

$$w_x^* = \begin{cases} 0 & \text{if } |\mathbf{R}^+| + |\mathbf{R}^-| < 3 \text{ or} \\ & |\mathbf{R}^-| = 0 \text{ or } |\mathbf{R}^+| = 0 \\ \alpha \dfrac{1}{\varepsilon + \mu_x^*} & \text{otherwise} \end{cases} \tag{11}$$

$$w_x = \begin{cases} 0 & \text{if } w_x^+ < w_x^* \\ w_x^+ - w_x^* & \text{otherwise} \end{cases} \tag{12}$$

where ε and α are positive constants, set in our experiments at 0.01 and 0.8, respectively. The term α has been introduced to prevent features found in both negative and positive examples from being discarded entirely.

Looking at these formulas, we observe that:

1. If there are at least three examples (of relevant, or non-relevant images) the weights are updated; otherwise they are all set at $1/\varepsilon$ by default.
2. If the user selects only relevant images, the weights are computed according to Eq. (10). For any given feature, the w_x^+ term is large when there is some form of agreement among the feature values of the selected images. We have already seen that treating all the relevant images in the same way may produce very poor results

when the images selected resemble the query image only in some pictorial features, but are actually quite different from it, and from each other. For any given feature the w_x^* term of Eq. (11), is large when there is some form of agreement among the feature values of positive and negative examples. This should mean that the feature is not discriminant for the query; consequently its weight is decreased [Eq. (12)].

4.3. Query Formulation

If the task is a target search, that is, to find specific images of which the user can supply a visual example or sketch, or a textual description, or both, the query vector representing the user's information needs must be preserved by taking into account the feature vectors of the images judged relevant by the user.

In all other cases, in which the user wants to retrieve images that cannot be fully represented by a single example, e.g. the user wants to retrieve all the images in which a forest scene is depicted, and/or which are mostly red, some form of query processing must be performed in order to better represent the user's information needs in a single query vector. In visual querying, one way of doing this is to take a weighted average of query feature vectors and of relevant images [18, 43]. But in the case of the forest scene, for example, the algorithm cannot provide for the fact that relevant images may differ from the original query with respect to some features. On the other hand, processing all the relevant images as single queries, and then combining the retrieval outputs may create an unacceptable computational burden when the database is large. Our approach is to let $\mathbf{R}^+$ be the set of relevant images the user has selected (including the original query), while $\bar{\mathbf{Q}}$ is the average query, and $\bar{\sigma}$, the corresponding standard deviation. We then proceed as follows:

$$\mathbf{Y}_b(j) = \{\mathbf{X}_b^i(j) \mid |\mathbf{X}_b^i(j) - \bar{\mathbf{Q}}_b(j)| \leq 3\bar{\sigma}_b(j)\} \quad \forall\ b, i, \text{and } j \tag{13}$$

$$\tilde{\mathbf{Q}}_b(j) = \frac{1}{|\mathbf{Y}_b(j)|} \sum_{X_b^i(j) \in \mathbf{Y}_b(j)} \mathbf{X}_b^i(j) \tag{14}$$

The query processing formulates a new visual query $\tilde{\mathbf{Q}}_b$ that better represents the images of interest to the user, taking into account the features of the relevant images, without allowing one different feature value to bias query computation.

A similar process is used for text: words found in relevant texts are added together to increase the weights associated with each word according to their relative frequency in the texts; instead, words present in both relevant and non-relevant texts are discarded. Again, letting $\mathbf{T}^+$ be the set of the relevant texts and $\mathbf{T}^-$, the set of non-relevant texts, the new textual query $\tilde{T}$ is computed as

$$\tilde{T} = \left(\bigcup_{T_i \in \mathbf{T}^+} T_i \right) \Big\backslash \left(\bigcup_{T_j \in \mathbf{T}^-} T_j \right) \tag{15}$$

4.4. Image Filtering

Since comparing a query Q with *every* image I in the database is a time-consuming task, we have implemented a method for filtering the database before the pictorial distances

are actually computed. This method is based on a variant of *triangle inequality* as proposed by Berman and Shapiro [4], and has the advantage of being applicable to any distance measure that satisfies triangle inequality.

We let I represent a database image, Q the query, K a reference image called *Key*, and d a distance measure. The following inequality holds for every Q, K and I:

$$d(I, Q) \geq |d(I, K) - d(Q, K)| \tag{16}$$

Assuming that we have precalculated $d(I, K)$ for all the images (I) in the database, to retrieve all the images such that distance $d(I, Q)$ is not greater than a threshold S, we can use the previous inequality to filter the database as follows:

1. compute query Q,
2. compute $d(Q, K)$,
3. find all the images (I) having $\alpha \leq d(I, K) \leq \beta$ where $\alpha = d(Q, K) - S$ and $\beta = d(Q, K) + S$.

Step 3 is a direct consequence of triangle inequality, adding the condition $d(I, Q) \leq S$. Since the distances $d(I, K)$ have already been calculated, we can store them directly in the database, and a standard SQL query can be used to retrieve the correct images.

If distance d is a linear combination of distances d_i, as in our case, we can apply triangle inequality to each term of the measure, adjusting the threshold of each inequality as follows. Assuming that $d(I, Q) = w_1 d_1(I, Q) + \cdots + w_i d_i(I, Q) + \cdots + w_n d_n(I, Q)$, since we want $d(I, Q) \leq S$, we have $w_i d_i(I, Q) \leq S$ for $i = 1, \ldots, n$, thus $d_i(I, Q) \leq S/w_i$. We have n conditions, similar to those in the previous case, that must be verified simultaneously, meaning that in the SQL query, they are and-ed together.

One image key K alone is not enough to discriminate the contents of the database, so m keys are used, and the results of each key filter are and-ed together to obtain the final results. The number of keys chosen, m, will be a function of N, the number of images in the database. A good compromise between a suitable number of keys and limited data storage space is provided by a logarithmic function that increases the number of keys slowly:

$$m = \log_{10} N \tag{17}$$

The method based on triangle inequality, performs better than a sequential search even when the keys are selected at random, as we have done. The threshold S is updated according to the weights determined by relevance feedback using the equation

$$S' = S \frac{MaxDist}{1/\varepsilon} \tag{18}$$

where *MaxDist* is the maximum value of the distance that can be obtained from the distance measure with the selected weights. The complete filtering method can be

summarized as follows:

1. compute query Q and weights w_i
2. compute threshold S'
3. compute $d_i(Q, K_k)$ for $i = 1, \ldots, n$ and $k = 1, \ldots, m$,
4. find all the images (I) such that
   $\alpha_{11} \leq d_1(I, K_1) \leq \beta_{11}$ and $\ldots$ and $\alpha_{1m} \leq d_1(I, K_m) \leq \beta_{1m}$ and
   $\ldots$
   $\alpha_{n1} \leq d_n(I, K_1) \leq \beta_{n1}$ and $\ldots$ and $\alpha_{nm} \leq d_n(I, K_m) \leq \beta_{nm}$
   with $\alpha_{ij} = d_i(Q, K_j) - S'$ and $\beta_{ij} = d_i(Q, K_j) + S'$,
5. compute the similarity $d(I, Q)$ on the remaining images and rank them.

5. Implementation and Results

The Quicklook[2] system (Figure 2) has been implemented in Visual C++. It is composed of three independent subsystems. The first, the indexing submodule, which indexes the pictorial content of the images and the available textual information. The second, the retrieval submodule, applies relevance feedback to retrieve the desired images from the database, once a query (visual and/or textual) has been submitted. The third subsystem, the manager module, contains all the supporting utilities. In general, the use of the system involves running all these modules.

When a new image database is fed into the system, the corresponding thumbnails for display are computed, and the corresponding textual information, if available, is also imported. Then, the visual and textual indices are computed (see Section 2). The default visual index contains: the ratio between the dimensions of the images, the color coherence vectors (CCV), the spatial chromatic histogram (SCH), the moments of inertia (mean, variance, skewness and kurtosis), the histogram of contour directions, the

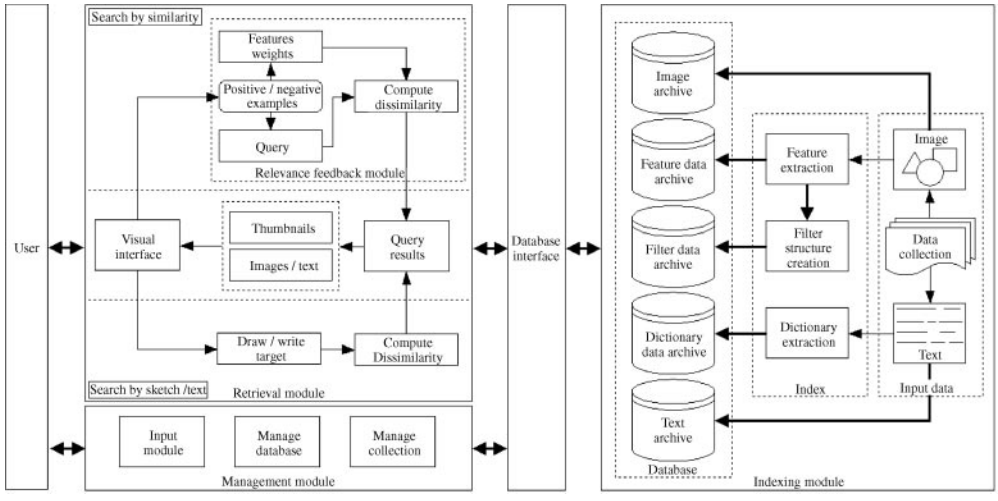


Figure 2. Quicklook[2] system architecture

mean and variance of the absolute values of the coefficients of the sub-images at the first three levels of the multi-resolution Daubechies wavelet transform of the luminance image, and the spatial composition of the color regions identified by the process of quantization in 11 colors. These can be calculated on the global image and/or on sub-images obtained by dividing the original image in different ways and then recomposing it, if desired (Figure 1). The indices may be updated by the user if new images are added to the database or the user wants to enlarge the set of features used for indexing. Once the images have been indexed, the filter data archive and the dictionary data archive are automatically computed and stored. At the beginning of a query session the user may modify the default retrieval strategy, which employs both visual and textual indices.

To avoid the time-consuming task of comparing a query Q with every image I in large databases, the user may apply the image filter to reduce the number of images that are retrieved before computing the distances. The user may also use standard SQL queries based on exact text matches to reduce the number of images selected for further querying (Figure 3).

To start a query session a user may:

1. provide the system an example ready-made (image and/or text) of his information needs;
2. sketch in an image and/or type in a text using the available tools;
3. browse the database to find one, or more relevant images with which to begin.

In the first two cases, the relevance feedback mechanism does not modify the query representing the user's information needs, but only the weights used in the evaluation of

Figure 3. Query setting interface

similarity. When a query is submitted, the system rearranges the database images in order of decreasing similarity with respect to the query, and then shows the user the most similar images. The user is allowed to browse textual annotations, and may, in successive iterations, mark any of the retrieved items as relevant, or non-relevant. A new query vector can then be computed, on the basis of the features of the relevant images, and the overall evaluation of the dissimilarity function updated, taking into account the features of both relevant and non-relevant images. There is no limit to the number of images that can be selected, or to the number of relevance feedback iterations. The user ends interaction with the system when he finds the desired images, or decides that they cannot be found because either the system is unable to decipher his information needs, or the desired images are not present in the database.

In order to quantify the improvement in image retrieval obtained by applying the relevance feedback mechanism, we have applied a measure called Effectiveness (efficiency of retrieval, or fill ratio), was applied here [13]. Let S be the number of images retrieved in the short list when posing a query; $\mathbf{R}_q^I$, the set of relevant images in the database; and $\mathbf{R}_q^E$, the set of images retrieved in the short list (considered 'relevant' by the system). The effectiveness measure is defined as

$$\eta_S = \begin{cases} \dfrac{|\mathbf{R}_q^I \cap \mathbf{R}_q^E|}{|\mathbf{R}_q^I|} & \text{if } |\mathbf{R}_q^I| \leq S \\ \dfrac{|\mathbf{R}_q^I \cap \mathbf{R}_q^E|}{|\mathbf{R}_q^E|} & \text{if } |\mathbf{R}_q^I| > S \end{cases} \tag{19}$$

If $|\mathbf{R}_q^I| \leq S$, the effectiveness is reduced to the traditional recall measure, while if $|\mathbf{R}_q^I| \leq S$, the effectiveness corresponds to precision.

The similarity retrieval features of the Quicklook[2] system has been tested on 15 different databases for a total of over 50,000 images. These databases were generated in the framework of feasibility studies of potential applications of the system, and include several collections of textiles, ceramics and trademarks, together with various archives of painting and photographs, both in color and in black and white. Some of the experimental results, quantitatively evaluated on thematic databases of up to 1800 images, can be found in references [28, 30, 33, 34, 44, 45]. Relevance feedback improves the effectiveness of the retrieval by 20–25% for all the databases. In general, the second iteration (the first relevance feedback iteration) corresponds to the largest single improvement. We have observed, on the contrary, little benefit in repeating the procedure for more than five or six times. This is probably due to the limited capability of the low-level features used to exhaustively describe the image content, and not to the mechanism itself. Since it could be argued that this performance could not be obtained on larger database, we have repeated the experiment on a photograph database of about 12,000 images, as shown below. No textual annotation has been used in indexing the images; however, all the pictorial features listed in Section 3 have been included in order to understand whether multiple representations of the same visual cues can truly and effectively integrated, and synergically used by the relevance feedback and query reformulation mechanisms. Each retrieval iteration takes 10 s on a Pentium III 550 Mhz.

In Table 1 we have summarized the experimental results for a total of 28 queries randomly chosen (Figure 4). The ground truth similarity was assessed on the basis of the

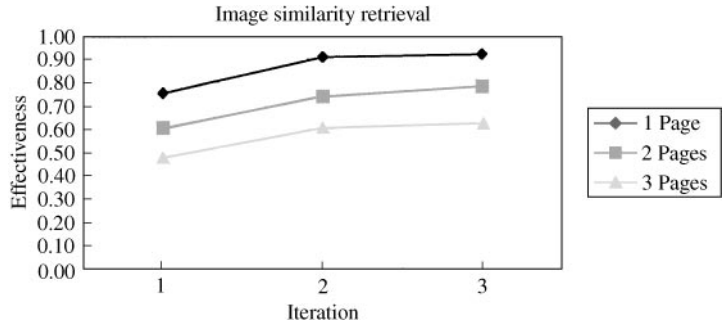


Table 1. Summary of the image similarity retrieval experiment

	1 Page			2 Pages			3 Pages		
Iteration	1	2	3	1	2	3	1	2	3
Min	0.50	0.54	0.61	0.34	0.32	0.46	0.26	0.31	0.40
Mean	0.76	0.91	0.92	0.61	0.74	0.78	0.48	0.61	0.62
Max	1.00	1.00	1.00	1.00	0.98	1.00	0.92	0.90	0.93

Figure 4. The 28 queries used to evaluate the image similarity search

image category label attached to each image by the database provider (e.g. playing cards, sunset, etc.). These were not used for retrieval but only to benchmark the results. In Table 1, for the sake of completeness, the minimum, average and maximum effectiveness value at each of the first three retrieval iterations are reported for three short lists of different lengths (28, 56, and 84, respectively, corresponding to 1, 2, and 3 pages). The graph illustrates the average effectiveness with respect to the relevance feedback iteration, evaluated on the three short lists: a steeper slope in the first interval with respect to the second can be noted. These results confirm that the relevance feedback and query processing mechanisms, that is, user interaction with the system, makes it possible to correlate visual features with the images' semantic contents. Figures 5 and 6 present some examples of the system's application.

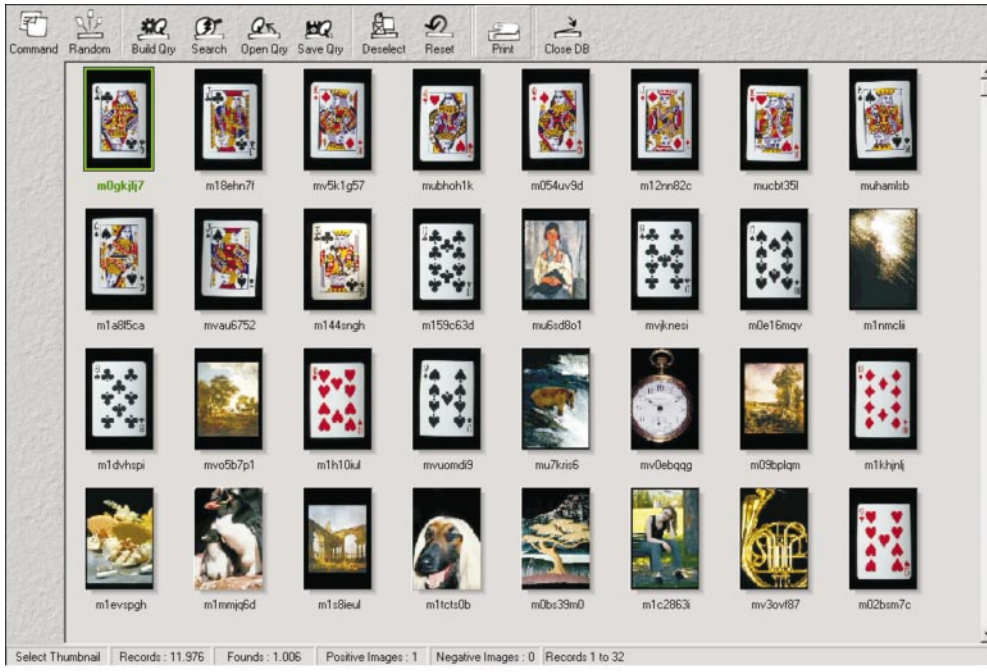
In Figure 5(a) and (b) a set of playing cards was searched. The top left image of Figure 5(a) is the query. Most of the images retrieved are playing cards but others which have color similar to the query also appeared. After selecting three of the retrieved images as relevant ('mvuomdi9', 'm02bsm7c', and 'm159c63d') all the images retrieved are of playing cards [Figure 5(b)]. Figure 6(a) shows the results of the search based on the hand drawn picture in the top left: all the images retrieved are similar in color to the query. Figure 6(b) shows the result after selecting two relevant images ('mv9bp4n5' and 'mvih8msr') and one non-relevant image ('mvpfs2s9').

We have also compared the performance of the indices (textual, visual, and the combination of both) in a target search framework, testing our system on a collection of 1732 paintings accompanied by textual cards providing a broad description of the subjects depicted in the image. The main reason for experimenting with a search for specific images rather than performing a similarity search is that the criteria applied are more objective. We submitted to a panel of four users a set of 30 target images each (Figure 7). After a preliminary phase in which users were allowed to familiarize themselves with the complete database, they were asked to retrieve each target image in turn. For each query the users performed three retrieval sessions, using first the textual index, then the visual index, and last the integration of both.

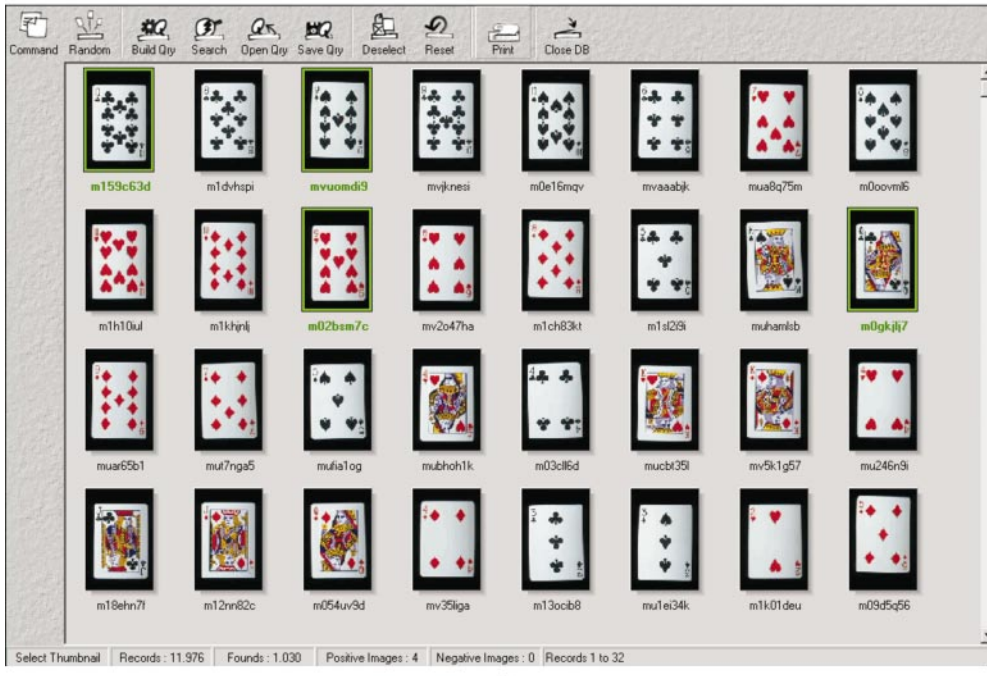
The retrieval sessions started with a random selection of 30 images; the user was allowed to select any number of relevant, or non-relevant images, but only from the first page, and then query the system. The system reordered the database images, the rank of the target image was taken as the retrieval score. If the desired images did not appear in the first set of images, the user was allowed to iterate the search selecting and/or discarding the relevant and/or non-relevant images, and then resubmit a new query: a limit of 10 iterations was allowed for each query. An image was considered successfully retrieved, if it appeared within the first display. The results of this experiments, averaged by the number of target images and by the number of users that participated in the experiments, are reported in Table 2, which gives the average rank of the target images for the ith iteration. The normalized results, with respect to the number of database images, are reported in the corresponding graph.

We note that:

- the integrated use of textual and visual indices improves the retrieval performance significantly;
- the use of relevance feedback significantly improves the system's performance no matter what indices are used. When it is applied, there is a rapid convergence to the



(a)



(b)

Figure 5. **(a)** Initial retrieval results **(b)** Retrieval results after the first iteration of relevance feedback

(a)

(b)

Figure 6. (a) Initial retrieval results **(b)** Retrieval results after the first iteration of relevance feedback

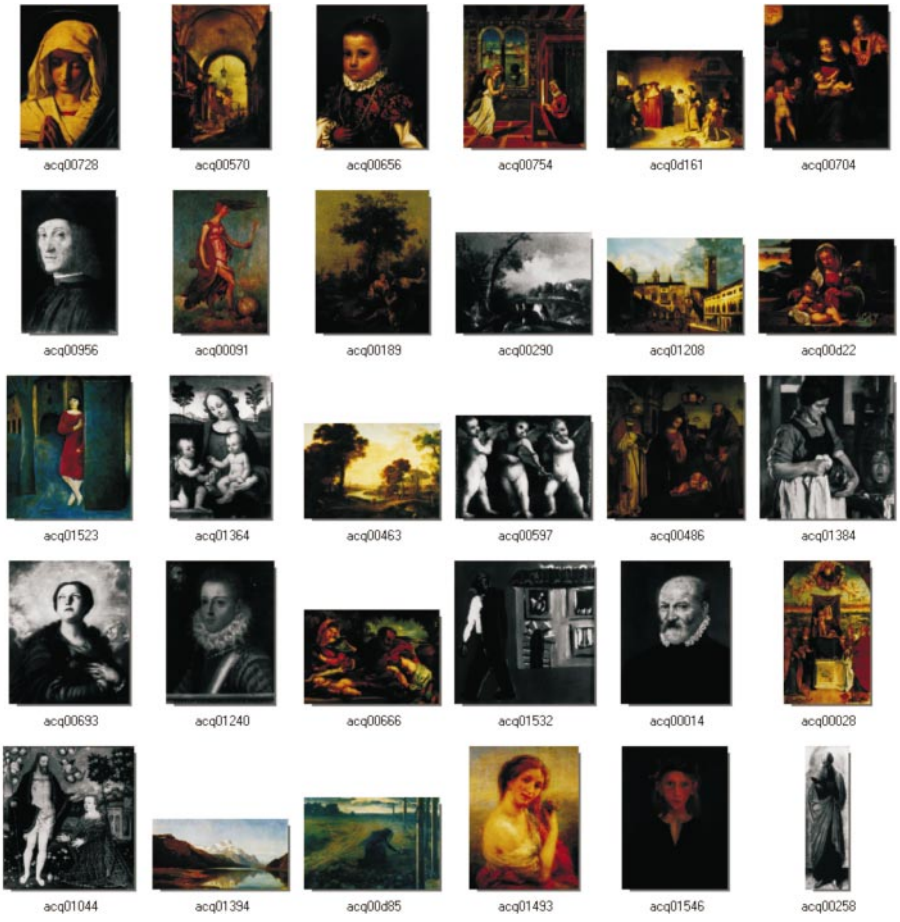


Figure 7. The 30 queries used to evaluate the target search experiments

desired images: for textual indices, visual and combined ones the iteration means are, respectively, 3.75, 4.42 and 2.42;

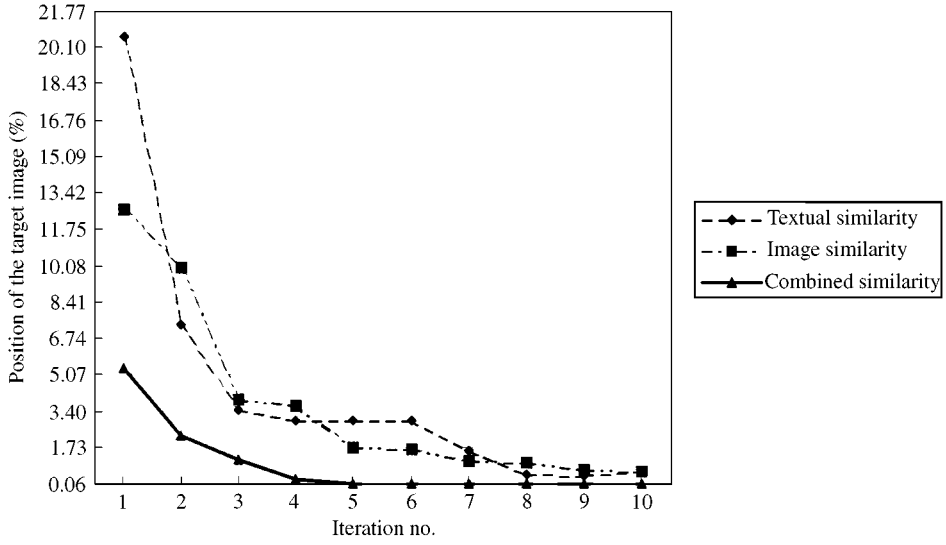
- the performance is slightly better for visual indices than for textual ones, confirming the feasibility and usefulness of content-based image retrieval, since visual indices are computed in a completely unsupervised manner by the system, while the textual ones are inevitably subjective, and depend upon the person compiling the annotations.

6. Conclusions and Future Work

We have described the main features of the multimedia information retrieval engine of Quicklook^2. Quicklook^2 allows the user to query image and multimedia databases with the aid of sample images, or a user-made sketch and/or textual descriptions, and progressively refine the system's response by indicating the relevance, or non-relevance of the retrieved items. The major contribution of this research is the design of

Table 2. Average rank for 30 queries and four users. In the corresponding graph the data in the table have been normalized with respect to the database size

Iteration	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Textual	356.42	128.00	59.67	51.67	51.42	50.92	27.75	8.42	6.92	10.00
Image	219.58	172.92	67.50	63.00	29.25	28.75	18.92	18.33	12.08	10.17
Combined	93.42	39.58	20.42	4.33	1.17	1.00	1.00	1.00	1.00	1.00



a mechanism that, through the statistical analysis of the image feature distributions and of the textual descriptions of the retrieved items the user has judged relevant, or non-relevant, identifies what features the user has taken into account (and to what extent) in formulating this judgement. It then modifies the contribution of the different visual and textual features in the overall evaluation of image similarity, as well as in the formulation of a single new query that better represents the user's information needs. The design of a content-based image retrieval system must address issues of efficiency in addition those of effectiveness. Another significant aspect of this study is the design and integration with the relevance feedback mechanism of an indexing scheme based on triangle inequality.

There are a number of open issues in the present implementation of the system which we plan to address in the early future:

1. The use of all the image features available and listed in this section limits the system's efficiency, but not its effectiveness and this is a problem already remedied. We are currently studying a friendly interface that will allow the user to inform the system of the type of database to be indexed; the system will then automatically discard (i.e. not compute) some pictorial features [46].
2. In the query by example mode the selection of the initial set of images to show to the user is critical when the database is large. At present Quicklook^2 offers

a database preview by random access, to find one, or more relevant images with which to begin. A database preview by clustering would be decidedly useful, but this is difficult as we have to take into account both visual and textual information to do so. This problem may be already circumvented to some extent in the current version of the system by using the query by sketch and/or query by text (i.e. a text digitized by the user) to obtain an initial set of images.

3. At the first retrieval iteration, when the user has selected just one image to search for, all the weights in the similarity function (9) are set at the value of $1/\varepsilon$. For faster tuning of the similarity function, the system could exploit previous query sessions performed by the user on the same database. To this end the user would be allowed to register satisfactory queries together with the corresponding weights in the similarity measure. When the user has already formulated a query 'similar' to the new one, the algorithm would then set the initial weights of the similarity function at the value of the former query, reducing the time and effort needed to adapt the similarity measure by means of the relevance feedback algorithm.
4. Although Quicklook^2 has several features that code local image characteristics, no segmentation or object recognition is performed. In this sense, Quicklook^2 has limited capabilities for discriminating images by their spatial contents. This feature could easily be added in the framework of particular application domains where the necessary information about the images to be indexed can be supplied.

References

1. O. Aigrain, H. Zhang & D. Petkovic (1996) Content-based representation and retrieval of visual media: a state-of-the-art review. *Multimedia Tools and Applications* **3,** 179–182.
2. V. N. Gudivada & V. V. Rahavan (1997) Modeling and retrieving images by content. *Information Processing and Management* **33,** 427–452.
3. A. Del Bimbo & R. Schettini (1998) Chairmen introduction. *Proceedings of the Image and Video Content-based Retrieval* (A. Del Bimbo, R. Schettini, eds), pp. i–iii.
4. A. P. Berman & L. G. Shapiro (1999) A flexible image database system for content-based retrieval. *Computer Vision and Image Understanding* **75,** 175–195.
5. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack & D. Petrovic (1994) Efficient and effective querying by image content. *Journal of Intelligent Systems* **3,** 231–262.
6. J. R. Bach, C. Fuller, A Gupta, A. Humpapur, H. Horowitz, R. Jain & C. Shu (1996) The Virage image search engine: an open framework for image management. *Proceedings of the SPIE Storage and Retrieval for Still Image and Video Database IV*, S. Jose.
7. W. Y. Ma & B. S. Manjunath (1996) Netra: a toolbox for navigating large image databases. *Proceedings of the IEEE International Conference on Image Processing.*
8. J. Feder (1996) Towards image content-based retrieval for the World-Wide Web. *Advanced Imaging* **11,** 26–29.
9. Y. Rui & T. S. Huang (1999) Image retrieval: current technologies, promising directions, and open issues. *Journal of Visual Communication and Image Representation* **10,** 39–62.
10. C. Nastar, M. Mitschke, C. Meilhac & N. Boujemaa (1998) Surfimage: a flexible content-based image retrieval system. *Proceedings ACM-Multimedia*, Bristol, England, September 12–16.
11. A. Pentland, R. Picard & S. Sclaroff (1996) Photobook: tools for content-based manipulation of image databases. *International Journal of Computer Vision* **18,** 233–254.
12. E. Binaghi, I. Gagliardi & R. Schettini (1994) Image retrieval using fuzzy evaluation of color similarity. *International Journal of Pattern Recognition and Artificial Intelligence* **8,** 945–968.

13. B. M. Mehtre, M. S. Kankanhalli & W. F. Lee (1998) Content-based image retrieval using a composite color-shape approach. *Information Processing and Management* **34,** 109–120.
14. A. F. Smeaton & I. Quigley (1996) Experiments on using semantic distances between words in image caption retrieval. In: *Proceedings of the 19th International Conference on Research and Development in Information Retrieval* (*SIGIR96*), Zurich, Switzerland.
15. Y. Zhuang, Y. Rui, T. S. Huang & S. Mehrotra (1998) Applying semantic association to support content-based video retrieval. *Proceedings of IEEE VLBV98 Workshop*, Urbana, IL, pp. 45–48.
16. V. Harmandas, M. Sanderson & M. D. Dunlop (1997) Image retrieval by hypertext links. *SIGIR 1997*, pp. 296–303.
17. S. Mukherjea & J. Cho (1999) Automatically determining semantics for Word Wide Web multimedia information Retrieval. *Journal of Visual Languages and Computing* **10,** 585–606.
18. M. La Cascia, S. Sethi & S. Sclaroff (1998) Combining textual and visual cues for content-based image retrieval on the world wide web. *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, June.
19. M. Ortega-Binderberg, S. Mehrotra, K. Chakrabarti & K. Porkaew (2000) WebMARS: a multimedia search engine. *Proceeding IS& T/ SPIE Conference on Internet Imaging*, San Jose, CA, SPIE, New York, Vol. 3964, January, pp. 314–321.
20. M. Agosti, F. Crestani & M. Melucci (1997) On the use of information retrieval techniques for the automatic construction of hypertext. *Information Processing and Management* **33,** 133–144.
21. J. Allan (1997) Building hypertext using information retrieval. *Information Processing and Management* **33,** 145–159.
22. G. Salton, A. Singhal, M. Mitra & C. Buckley (1997) Automatic text structuring and summarization. *Information Processing and Management* **33,** 193–207.
23. D. Tudhope & Taylor (1997) Navigation via similarity: automatic linking based on semantic closeness. *Information Processing and Management* **33,** 233–242.
24. M. Agosti, J. Allan (eds.) (1997) Special Issue on Methods and Tools for the Automatic construction of Hypertext. *Information Processing and Management* **33** (2).
25. T. Minka & R. W. Picard (1997) Interactive learning with a "Society of Models". *Pattern Recognition* **30,** 565–581.
26. I. J. Cox, M. L. Miller, S. O. Omohundro & P. N. Yianilos (1996) PicHunter: Bayesian relevance feedback for image retrieval. *Proceedings of the ICPR'96*, pp. 361–369.
27. Y. Rui, T. S. Huang, M. Ortega & S. Mehrotra (1998) Relevance feedback: a power tool in interactive content-based retrieval. *IEEE Transaction on Circuits and Systems for Video Technologies* (Special Issue on Interactive Multimedia Systems for the Internet) **8,** 644–655.
28. G. Ciocca & R. Schettini (1999) A relevance feedback mechanism for content-based image retrieval. *Information Processing and Management* **35,** 605–632.
29. G. Pass, R. Zabih & J. Miller (1996) Comparing images using color coherence vectors. *Proceedings of the Fourth ACM Multimedia 96 Conference*, Boston, MA, U.S.A.
30. G. Ciocca, I. Gagliardi & R. Schettini (1998) Retrieving color images by content. In: *Proceedings of the Image and Video Content-Based Retrieval Workshop*, Milan, Italy. (A. Del Bimbo, R. Schettini, eds), pp. 57–64.
31. L. Cinque, S. Levialdi & A. Pellicano (1999) Color-based image retrieval using spatial-chromatic histograms. *IEEE Multimedia Systems 99*. IEEE Computer Society, Silverspring, MD, Vol. II, pp. 969–973.
32. M. Stricker & M. Orengo (1995) Similarity of color images. *Proceedings of the SPIE Storage and Retrieval for Image and Video Databases III Conference*, San Diego, CA, U.S.A.
33. G. Ciocca & R. Schettini (2000) Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recognition* (accepted March 2000, in print).
34. R. Schettini, G. Ciocca & I. Gagliardi (2000) Interactive visual information retrieval. *Proceedings of the 2000 IEEE International Symposium on Circuits and Systems*, 28–31 May (Special Session on Digital Photography **V**, pp. 109–112.
35. M. Hu (1962) Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory* **8,** 179–187.
36. M. Amadasun & R. King (1989) Textural features corresponding to textural properties. *IEEE Transaction on System*, *Man and Cybernetics* **19,** 1264–1274.

37. H. Tamura, S. Mori & T. Yamawaki (1978) Textural features corresponding to visual perception. *IEEE Transactions on System*, *Man and Cybernetics* **8,** 460–473.
38. Y. Miyake, H. Saitoh, H. Yaguchi & N. Tsukada (1990) Facial pattern detection and color correction from television picture for newspaper printing. *Journal of Imaging Technology* **16,** 165–169.
39. P. J. Rousseeuw & A. M. Leroy (1987) *Robust Regression and Outlier Detection*. John Wiley & Sons, New York.
40. G. Salton (1989) *Automatic Text Processing*. Addison-Wesley, New York.
41. C. Cacciari (ed) (1995) *Similarity in language*, *thought and perception*, Brepols, Brussels.
42. A. M. Mood, F. A. Graybill & D. C. Boes (1988) *Introduzione alla statistica*. McGraw-Hill, New York.
43. M. Mitra, J. Huang & S. R. Kumar (1997) Combining supervised learning with color correlograms for content-based image retrieval. *Proceedings of the 5th ACM Multimedia 97 Conference*.
44. G. Ciocca, I. Gagliardi & R. Schettini (1999) Content-based color image retrieval with relevance feedback. *Proceedings of the International Conference on Image Processing*, Kobe (Japan) (Special session "Image Processing Based on Color Science").
45. G. Ciocca, I. Gagliardi & R. Schettini (1999) Quicklook: a content-based image retrieval system with learning capabilities. *IEEE Multimedia Systems 99*. IEEE Computer Society, Silverspring, MD, Vol. II, pp. 1028–1029.
46. C. Brambilla, I. Gagliardi, R. Schettini & A. Valsasna (2000) Detecting feature importance via tree classifiers—an experience. In: *Data Mining II* (N. Ebecken, C. A. Brebbia, eds.). Witt Press, Southampton, Boston, U.S.A., pp. 487–493.