The QuickLook image search engine

G. Ciocca, R. Schettini*

Istituto Tecnologie Informatiche Multimediali
Consiglio Nazionale delle Ricerche
Via Ampere 56, 20131 Milano, Italy
e-mail: [ciocca, schettini]@itim.mi.cnr.it

Abstract: This paper describes the main features of our image search engine: QuickLook. QuickLook allows the user to query image and video databases with the aid of example images or a user-made sketch, and progressively refine the system's response by indicating the relevance, or non-relevance of the retrieved items.

Keywords: image database, content-based image retrieval, image indexing, relevance feedback.

1. Introduction

Visual information systems operate on multimedia databases to recover relevant images and videos in response to a query. The systems currently employed, mainly support the search and the retrieval of pictorial information in two ways: with text-based keywords, or on the basis of low-level image features (color, texture, shape, spatial layout, ...) that must then be compared on the basis of similarity measures that are defined interactively by the user [1,2,3].

While using these approaches to retrieve a suitable image from an archive is often an inefficient and time consuming business, we have observed that users do not find it difficult to provide examples of similar and dissimilar images interactively. We have exploited this capacity in developing the image search engine of QuickLook, which allows the user to query image and video databases with the aid of example images or a user-made sketch, and progressively refine the system's response by indicating the relevance, or non-relevance of the retrieved items.

2. Image Indexing

Because perception is subjective, there is no one "best" representation of image content. The features listed below constitute a general purpose library of low-level features which can be calculated on the global image and/or sub-images obtained by dividing the original image in different ways:

- the ratio between the dimensions of the images;
- the Color Coherence Vectors (CCV) and Color Histogram in the CIELAB color space quantized in 64 colors [4];
- a histogram of the transition in colors (using a CIELAB color space quantized in 11 colors, namely red, orange, yellow, green, blue, purple, pink, brown, black, grey and white) [5];
- the Spatial Chromatic Histogram (SCH), synthesizing information about the location of pixels of similar color and their arrangement within the image [6];
- the moments of inertia (mean, variance, skewness and kurtosis) of the color distribution in the CIELAB space [7];
- a histogram of opportunely filtered contour directions (only high gradient pixels are considered). Edges are extracted by Canny's edge detectors, and the corresponding edge directions are quantized in 72 bins at 2.5° intervals. To compensate for different image sizes, the histograms are normalized with respect to the total number of edge pixels detected in the image [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 [9];
- the Neighborhood Gray-Tone Difference Matrix (NGTDM), i.e. coarseness, contrast, busyness, complexity, and strength, as proposed by Amadasum and King [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 [11].

*Corresponding Author.
This project is supported by the CNR Progetto Finalizzato 'Materiali Speciali per Tecnologie Avanzate II'

The total number of features is rather high since the color and direction histograms used in indexing are intrinsically large. However some of these features are not calculated, but derived from others during matching. All the features, except the SCH features which are compared using the distances metric proposed in [6], are compared with the L_1 distance measure, as it is statistically more robust than the L_2 distance measure [12].

3. Relevance Feedback

Sub-vectors of features are indicated by X^i_h, where i is the vector index, and h the index of the feature. We indicate with D_h the distance associated with the feature h-th. The global metric used to evaluate the similarity between two images of the database is, in general, a linear combination of the distances between the individual features:

$$\mathrm{Dist}(X^i, X^j) = \sum_{h=1}^{p} w_h D_h(X^i_h, X^j_h) \qquad (1)$$

in which the w_h are weights. There are two drawbacks to this formulation of image similarity. First, since the single distances may be defined on intervals of widely varying values, they must be normalized to a common interval so that equal emphasis is placed on every feature score. Second, 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. Moreover, image similarity is user- and task- dependent [13], and this dependence is still not understood well enough to permit careful, a-priori selection of the optimal measure.

3.1 Normalization of Features

To cope with the problem of distances defined on different intervals of values, we use the following normalization derived from the Gaussian normalization [8, 14, 15]:

$$D(X^i, X^j) = \left[\frac{D_1(X^i_1, X^j_1)}{\mu_1 + K\sigma_1}, \ldots, \frac{D_h(X^i_h, X^j_h)}{\mu_h + K\sigma_h}, \ldots, \frac{D_p(X^i_p, X^j_p)}{\mu_p + K\sigma_p}\right]^T \qquad (2)$$
$$= \left[d_1(X^i_1, X^j_1), \ldots, d_h(X^i_h, X^j_h), \ldots, d_p(X^i_p, X^j_p)\right]^T$$

K is a positive constant that influence the number of out-of-range values: in our experiment 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 similarity function has the following form:

$$\mathrm{Dist}(X^i, X^j) = \sum^{p} w_h d_h(X^i_h, X^j_h) \qquad (3)$$

3.2 Estimation of Weights

We let $\mathbf{R^+}$ be the set of relevant images selected by the user ($\mathbf{R^+}$ is usually only an approximation of the set of images relevant to the query in the whole database); $\mathbf{d^+_h}$, the set of normalized distances (computed on the feature h) among the elements of $\mathbf{R^+}$; and μ^+_h, the mean of the values of $\mathbf{d^+_h}$. Similarly, we define $\mathbf{R^-}$ as the set of non relevant images selected by the user to serve as negative examples, while $\mathbf{d^-_h}$ is the corresponding set of distances. From $\mathbf{R^+}$ and $\mathbf{R^-}$ we are then able to determine whether the influence of a feature must be limited in computing the dissimilarity by reducing the corresponding weight: let $\mathbf{R^{+-}}$ be the union of $\mathbf{R^+}$ with $\mathbf{R^-}$, and $\mathbf{d^{+-}_h}$, the corresponding set of distances among its elements. Since we can not make any assumptions about the statistical distribution of the features of non-relevant images by analyzing $\mathbf{R^-}$ (the selected non-relevant images may be not representative of all the non-relevant images in the database), we exclude set $\mathbf{d^-_h}$ from $\mathbf{d^{+-}_h}$, obtaining a new set of distances: $\mathbf{d^*_h} = \mathbf{d^{+-}_h} \backslash \mathbf{d^-_h}$. If we let μ^*_h be the mean of the elements in $\mathbf{d^*_h}$, we can now determine the weight terms to use in Equation (3) as follows:

$$w^+_h = \begin{cases} \dfrac{1}{\varepsilon} & \text{if } |R^+| < 3 \\ \dfrac{1}{\varepsilon + \mu^+_h} & \text{otherwise} \end{cases} \qquad (4)$$

$$w^*_h = \begin{cases} 0 & \text{if } |R^+| + |R^-| < 3 \text{ or } |R^-| = 0 \text{ or } |R^+| = 0 \\ \dfrac{1}{\varepsilon + \mu^*_h} & \text{otherwise} \end{cases} \qquad (5)$$

$$w_h = \begin{cases} 0 & \text{if } w^+_h < w^*_h \\ w^+_h - w^*_h & \text{otherwise} \end{cases} \qquad (6)$$

where ε is a positive constant (set at 0.01 in our experiments). Looking at these formulas, we observe that:

- if there are at least three examples (of relevant or non-relevant images) the weights are updated; otherwise they are all set at $1/\varepsilon$.
- if the user selects only relevant images, the weights are computed according to Equation (4). For any given feature, the w^+_h 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 relevant images selected resemble the query image only in some pictorial features, but are actually quite different from each other [14].
- for any given feature the w^*_h term of Equation (5), 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 the corresponding weight is decreased (Equation 6).

The structure of the relevance feedback mechanism is entirely description-independent, that is, the index can be modified, or extended to include other features without requiring any change in the algorithm.

3.3 Query Processing

Query processing consists in modifying the feature vector of the query by taking into account the feature vectors of the images judged relevant by the user. One way of doing this is to take a weighted average of the query feature vectors and of the relevant images [14]. But in this case, the algorithm can not provide for the fact that relevant images may differ from the original query with respect to some features. Our approach is to let $\mathbf{R}^{+}$ be the set of relevant images the user has selected (including the original query), while $\overline{\mathbf{Q}}$ is the average query, and $\bar{\sigma}$, the corresponding standard deviation. We then proceed as follows:

$$\mathbf{Y}_h(j) = \left\{ X_h^i(j) \;\middle|\; \left| X_h^i(j) - \overline{Q}_h(j) \right| \leq 3\bar{\sigma}_h(j) \right\} \forall\, h, i, \text{ and } j \tag{7}$$

$$\tilde{Q}_h(j) = \frac{1}{\left|\mathbf{Y}_h(j)\right|} \sum_{X_h^i(j) \in \mathbf{Y}_h(j)} X_h^i(j) \tag{8}$$

The query processing formulates a new query $\tilde{\mathbf{Q}}_h$ 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.

The query process could be similarly applied to compute a query representing non-relevant examples. This seems of little practical interest as non-relevant examples are usually not similar to each other, and are, consequently, scattered throughout the feature space.

4. The System at Work

A full description of all the characteristics of QuickLook is beyond the scope of the paper; we describe here only those of image search engine. QuickLook, allows the user to query the database, using keywords (non described here), example images, or a user-made sketch.

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. QuickLook offers a database preview by random access, or image clustering to allow the user to find one, or more relevant images with which to begin. At the first retrieval iteration, when the user has selected just one image to search for, all the weights in the similarity function (3) are set at the value of $1/\varepsilon$. For faster tuning of the similarity function, the system can exploit previous query sessions performed by the user on the same database. To this end the user is 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 sets 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.

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. In subsequent iterations the user may mark any of the retrieved images as relevant, or not relevant. A new query vector is then computed, on the basis of the features of the relevant images, and the overall evaluation of the dissimilarity function is 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 and to the number of relevance feedback iterations. The user ends interaction with the system when he finds the desired images, or decides that they can not be found because either the system is unable to decipher his information needs, or the desired images are not present in the database.

Since comparing a query Q with *every* image I in the database may be a time-consuming task, we have implemented a method for filtering large databases before computing the distances. This method is based on a variant of the *triangle inequality approach* proposed by Berman and Shapiro [9].

The similarity retrieval features of the Quicklook 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. Relevance feedback improves the effectiveness of the retrieval considerably for all the databases by over 30%. In general, the second iteration (that is the first relevance feedback iteration) corresponds to the largest single improvement. We have observed, to the contrary, little benefit in repeating the procedure for more than five or six times. It can reasonably be argued that this is due to the limited capability of the low-level features used to exhaustively describe the image content, and not to the mechanism itself.

Fig. 1 present the system interface, Fig. 2 and Fig. 3 present an example of the system's application to a database of some 12,000 images. Additional examples may be found at the following address: http://www.itim.mi.cnr.it/Linee/Linea1/Sottolinea3/relfeme.htm

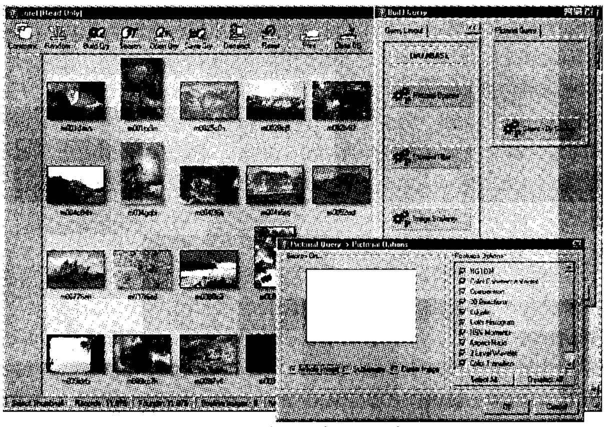

Figure 1. The new QuickLook interface. On the left are visible the query builder window and the query options window.

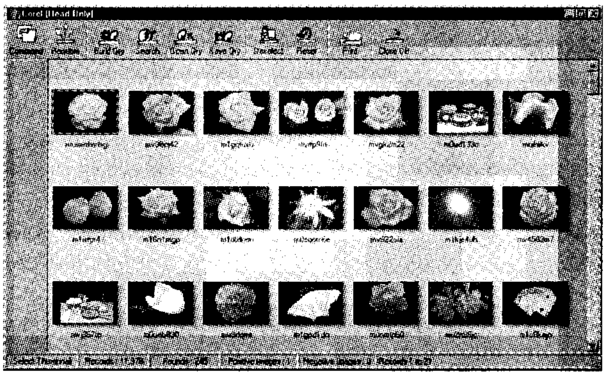

Figure 2. Example of retrieval results. The query is the top left image. No relevance feedback has been applied.

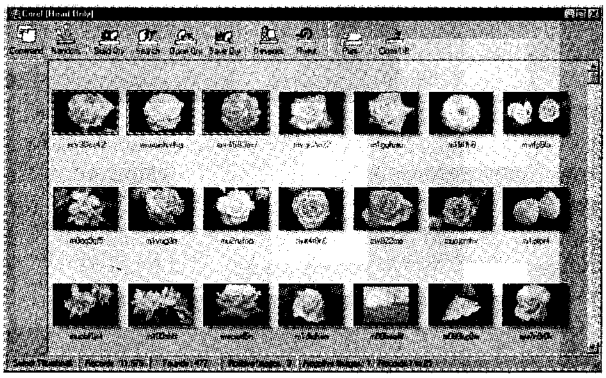

Figure 3. Retrieval results after the first iteration of relevance feedback. The last image of Figure 2 was selected as non relevant.

Reference

[1] Aigrain O., Zhang H., Petkovic D.: Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review. Multimedia Tools and Applications, 3 (1996) 179-182.

[2] Faloutsos C., Barber R., Flickner M., Hafner J., Niblack W., Petrovic D. Efficient and effective querying by image content, Journal of Intelligent Systems, 3 (1994) 231-262.

[3] Gudivada V.N, Rahavan V.V.: Modeling and retrieving images by content. Information Processing and Management, 33 (1997) 427-452.

[4] Pass G., Zabih R., Miller J.: Comparing Images Using Color Coherence Vectors. Proc. Fourth ACM Multimedia 96 Conference (1996).

[5] Gagliardi I., Schettini R.: A method for the automatic indexing of color images for effective image retrieval. The New Review of Hypermedia and Multimedia, 3 (1997), 201-224.

[6] Cinque L., Levialdi S., and Pellicano' A., Color-Based Image Retrieval Using Spatial-Chromatic Histograms, IEEE Multimedia Systems 99, IEEE Computer Society, Vol. II, (1999) 969-973.

[7] Stricker M., Orengo M.: Similarity of Color Images. Proc. SPIE Storage and Retrieval for Image and Video Databases III Conference (1995).

[8] Ciocca G., Gagliardi I., Schettini R., Content-based color image retrieval with relevance feedback, Proc. International Conference on Image Processing, Kobe (Japan), Special session "Image Processing Based on Color Science", (1999).

[9] Berman A.P., Shapiro L.G., A flexible image database system for content-based retrieval. Computer Vision and Image Understanding, Vol. 75, Nos. 1/2, July/August, (1999) 175-195.

[10] Amadasun M., King R.: Textural features corresponding to textural properties, IEEE Transaction on System, Man and Cybernetics, 19 (1989) 1264-1274.

[11] Ciocca G., Gagliardi I., Schettini R.: Retrieving color images by content. In: Del Bimbo A., Schettini R. (eds.) Proc. of the Image and Video Content-Based Retrieval Workshop (1998).

[12] Rousseeuw P.J., Leroy A.M.: Robust regression and outlier detection, John Wiley & Sons (1987).

[13] Binaghi E., Gagliardi I., Schettini R.: Image retrieval using fuzzy evaluation of color similarity. International Journal of Pattern Recognition and Artificial Intelligence, 8 (1994) 945-968.

[14] Ciocca G., R. Schettini, A relevance feedback mechanism for content-based image retrieval, Information Processing and Management, 35 (1999) 605-632.

[15] Mood A.M, Graybill F.A., & Boes D.C: Introduzione alla statistica. McGraw-Hill (1988)