Color Retrieval in Vector Space Model

Anca Doloc-Mihu[1], Vijay V. Raghavan[1], Peter Bollmann-Sdorra[2]

[1] The Center of Advanced Computer Studies,
University of Louisiana at Lafayette,
Lafayette, LA 70504, USA,
{axd9917, raghavan}@cacs.louisiana.edu
[2] Department of Computer Science,
Technical University of Berlin,
FR5-11, Franklin St. 28/29, D-1000, Berlin 10, Germany,
bollmann@cs.tu-berlin.de

Abstract. Many applications involving similarity search use the *QBIC* Euclidian distance to match two color histograms. To alleviate certain problems associated with this approach, which is based on a distance metric, in this paper, we propose a *Color-Color Similarity Retrieval Approach* to compute the similarities between images. This approach, based on the similarity matrix between feature vectors, leads to three new color-color similarity retrieval models, called *(PA, Q), (P, QA)*, and *(PB, QB)* if the color-color similarity matrix A is positive definite. By precomputing the similarity matrix and all the products *PA*, *PB*, *QA* and *QB*, where *P* and *Q* are respectively, vectors representing images from the database and *B* is obtained by decomposing *A* in a special way, the retrieval function becomes linear during the retrieval step. To compute the similarity matrix A, we propose a more general form than that of the *QBIC* approach.

In addition, in this paper, we introduce a new algorithm, *Kernel Rocchio Algorithm*, which combines the simplicity of *Rocchio* method with the power of non-linear kernel functions to improve the relevance feedback process. In this context, we prove that the proposed retrieval models are equivalent, in the sense that the query learned via relevance feedback in one model can also be learned in any of the other models.

We implement our algorithms and test them on a synthetic dataset that allows easy mechanism for specification of image relevance for a user query. For learning purpose, we also consider a model that we refer to as the *(P, Q)* model, which does not require the use of the matrix A. Our results show that the *(P, Q)* retrieval model, used together with the polynomial kernel, provides better results compared to other combinations of retrieval models and kernel functions. We believe that if the method of computing color correlations is improved, the similarity retrieval model (*(PA, Q), (P, QA)*, or *(PB, QB)*) should perform, quality-wise, just as well as the *(P, Q)* model, as shown by our theoretical results.

1 Introduction

In the *QBIC* system (Flickner et al., 1995) the distance between two images, P and Q, is computed by using a generalized Euclidian distance $\delta(P,Q) = (P-Q)^t A(P-Q)$, where the $N \times N$ matrix A defines the similarities between colors, and P and Q represents the color image histogram and the color query histogram, respectively, in the *RGB* color space. For many applications involving similarity search, this distance between feature vectors has certain drawbacks. First, images are represented as high-dimensional vectors such as histograms and thus, it is time consuming to use a quadratic function to compute the distances between them during retrieval. Second, several feedback algorithms such as Rocchio's relevance feedback (Rocchio, 1971), Perceptron (Wong et al., 1988), and Support Vector Machine (Vapnik, 1995) cannot be applied directly on distances.

In this paper, we propose a *Color-Color Similarity Retrieval Approach* to compute the similarities between images. Whereas common retrieval functions used in Information Retrieval, for document retrieval, ignore the correlations between features, our proposed similarity - based retrieval approach fulfill this requirement leading to three new similarity retrieval models. The new similarity function includes the feature similarity matrix A and is defined as following:

$$F : \mathbb{R}^{\mathbb{N}} \times \mathbb{R}^{\mathbb{N}} \rightarrow \mathbb{R}, F(P, Q) = P^t AQ,$$

where $A = (a_{ij})_{i,j=1,...,N}$, is a symmetric similarity matrix with a_{ij} being the similarity between features i, j. We assume $a_{ii} = 1$ and $0 \leq a_{ij} < 1$ for $i \neq j$. For example, matrix A can be computed as specified in the *QBIC* system. By using the associativity property of matrix multiplication on the proposed form of retrieval we obtain three new retrieval models, *(PA, Q)*, *(P, QA)*, and *(PB, QB)* if the similarity matrix is positive definite. For a positive definite matrix A, we can use *Cholesky decomposition*, i.e. there exists a unique matrix B such that $A = B^t B$, and we obtain a new form *(PB, QB)* for our retrieval purpose. By precomputing the similarity matrix and all the products *PA*, *PB*, *QA* and *QB*, the retrieval function becomes linear during the retrieval step.

Recently, many retrieval systems incorporate a learning method. However, to match two color histograms, most of them (Bimbo, 2001) use the above generalized Euclidian distance *(QBIC*, *Blobworld*, *MetaSeek*, *WebSeek)*, or a weighted form of it (*FIR*, *MARS*, *NETRA*). Some other systems use different metrics (like *L1)* for the same purpose (Veltkamp and Tanase, 2000). Instead, in this paper, we seek to take advantage of the proposed linear color retrieval models for searching an image database. One of the advantages of a linear retrieval function is that there are several training methods of feedback available, such as Rocchio (Rocchio, 1971), Perceptron (Wong et al., 1988) or the Support Vector Machine (Vapnik, 1995). In this context, we prove that the proposed retrieval models are equivalent, in the sense that the query learned in one model can also be learned in any of the other models.

To compute the similarity matrix A, we propose a more general form than that of the *QBIC* approach. That is, $A = (a_{ij})_{i,j=1,...,N}, a_{ij} = \begin{cases} 1 & i = j, \\ \tau f(d_{ij}) & i \neq j, \end{cases}$, where $\tau > 0$ is a constant, and $f : R^+ \rightarrow R^+$ is a strictly monotonic decreasing function depending on the distances between colors d_{ij}, with $0 \leq f(d_{ij}) < 1$ (the distance between two *RGB* colors $c_i = (r_i, g_i, b_i)$ and $c_j = (r_j, g_j, b_j)$ is given by $d_{ij} = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}$). We notice that in *QBIC* system the distances between two features (or, colors) is not the same as the distance between two images such that each image consists of only one of the two features, which may result in wrong ranking results. We prove that our similarity approach overcomes this problem.

In addition, in this paper we introduce a new algorithm, *Kernel Rocchio Algorithm*, which combines the simplicity of Rocchio method with the power of non-linear kernel functions to improve the relevance feedback process.

We implement our algorithms and test them on a synthetic dataset that allows easy mechanism for specification of image relevance for a user query. For learning purpose, we also consider a model that we refer to as the *(P, Q)* model, which does not require the use of the matrix A. Our results show that the *(P, Q)* retrieval model used together with the polynomial kernel provides better results compared to other combinations of retrieval models and kernel functions. We think the reason for this unexpected result is that the way of computing the similarity matrix based on the Euclidian distance between colors is inappropriate. If the method of computing color correlations is improved, the similarity retrieval model (*(PA, Q)*, *(P, QA)*, or *(PB, QB)*) should perform just as well as the *(P, Q)* model, as shown by our theoretical results.

The rest of the paper is organized as follows. In Section 2, we introduce important concepts used in this paper and we present our motivation. Our proposed *Color-Color Similarity Retrieval Approach* is described in Section 3. Section 4 deals with the learning process. This section includes also a new relevance feedback

algorithm, the *Kernel Rocchio Algorithm*. In Section 5, we provide the results from the experimental evaluation. Finally, Section 6 concludes the paper.

2 Motivation

In this section we build the background for our work and present our motivations for improving an existing approach used in the *QBIC* image retrieval system.

2.1 Image Representation in Color Space

In image retrieval, low-level features of images are extracted during the preprocessing step and then, they are used as a representative for the image to be matched against the query. Feature vectors represent the image projection onto the selected N-dimensional feature space. For example, for color representation, there are many color spaces used to represent the image (*RGB*, *HSV*, *CIE*, etc.), and many approaches used for representation, such as histograms and binary sets (Bimbo, 2001; Smith, 1997), correlograms (Smeulders et al., 2000).

In the sequel we consider an image as being represented by only its color histogram in the *RGB* space. Thus, each color c_i which appears within the image is represented as a 3D point in the *RGB* space, $c_i = (r_i, g_i, b_i)$, called the color bin.

The image histogram represents the distribution of the feature elements or the color bins within the image. It holds the frequency counts of the number of color bins of the image $P = (p_1, p_2, \ldots, p_N)$, where p_i is the frequency of occurrence of color bin c_i.

In this paper we represent the image by using histograms in the *RGB* color space, but our model can be applied to the other color spaces like *HSV, Lab* etc., maybe with some adjustements according to the characteristics of the chosen space. For convenience, across the paper we will use both terms image features (or features, for short) and color bins (or colors, for short) interchangeably. Color vectors in the *RGB* space are used to obtain the color-color similarity matrix in the next section.

2.2 Color-Color Similarity Matrix

Suppose we have a N-dimensional feature space $X = \{F_1, F_2, \ldots, F_N\}$. Given any two features from the same vector space X, F_i and F_j, we define the distance between them, as $d_{ij} = d(F_i, F_j)$.

We assume that a distance between features always exists. This assumption is made based on a different vector space used to represent these features. For example, in *RGB* color space the distance between two color bins $c_i = (r_i, g_i, b_i)$ and $c_j = (r_j, g_j, b_j)$ is given by

$$d_{ij} = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}\,. \tag{1}$$

That is, the d_{ij} distances form a (color-color) feature distance matrix $D = (d_{ij})_{i,j=1,\ldots,N}$. A few different methods are available to obtain a similarity matrix $A = (a_{ij})_{i,j=1,\ldots,N}$ by transforming the distances into similarities (Smith, 1997; Hafner et al., 1995). Next section describes briefly the similarity matrix and its usage as encountered in the *QBIC* retrieval system.

2.3 *QBIC* system

In the *QBIC* system (Flickner et al., 1995):

$$a_{ij} = 1 - \frac{d_{ij}}{M},\ \text{with}\ M = \max_{i,j}(d_{ij})$$

defines the similarity between the two colors c_i and c_j.

In the *QBIC* system (Flickner et al., 1995) the distance between two images, P and Q, is computed by using a quadratic generalized Euclidian distance:

$$\delta(P, Q) = (P - Q)^t \cdot A \cdot (P - Q)\,, \tag{2}$$

where the $N \times N$ matrix A defines the similarities between colors, and P and Q represents the color image histogram and the color query histogram, respectively, in the *RGB* color space. The following subsection presents some drawbacks of the *QBIC* distance between images (see Equation (2)).

2.4 Discussion on the *QBIC* distance

The *QBIC* distance between feature vectors has several disadvantages. First, images are represented as high-dimensional vectors such as histograms (e.g. 256 color bins) and thus, it is time consuming to use a quadratic function to compute the distances between them during retrieval. Second, in Information Retrieval there are several feedback algorithms, such as Rocchio's relevance feedback algorithm (Rocchio, 1971), Perceptron (Wong et al., 1988), and Support Vector Machine (Vapnik, 1995), that cannot be applied directly on distances.

Finally, the distance between two colors F_i and F_j, $d(F_i, F_j) = d_{ij}$, is not the same as the distance between the two single-color images P_i and P_j which contain only colors F_i and F_j, respectively, $\delta(P_i, P_j) = \frac{2}{M} d_{ij} \neq d_{ij}$ with M defined in Section 2.3. We do not see any natural reason why the distances between features and images that consist just of these features should be different. We call this the *distance mismatch* problem. Further, these differences between distances may result in wrong ranking results. Our *Color-Color Similarity Retrieval Approach*, presented in the next section, does not present these problems of the *QBIC* distance.

3 Proposed Retrieval Models to Achieve Linear Retrieval Function

In this section, we describe and analyse a similarity-based retrieval approach (*Color-Color Similarity Retrieval Approach*) that deals with the above shortcomings of the *QBIC* distance function.

3.1 Similarity Retrieval Approach

In Information Retrieval one of the most used model is the Vector Space Model, where documents and queries are represented as vectors. Then, a linear retrieval form is used in order to match the query vector agains documents from collection. In Image Retrieval the tendency is the same as in the Information Retrieval: to use the Vector Space Model by representing images by vectors of features. A good reason to use vector representations is that vector spaces are already well studied and there is a good theory behind which allows us to easily perform computations on them.

By analogy with document vector representation from Information Retrieval (Raghavan and Wong, 1986), we represent images (their features) as vectors in Image Retrieval. Suppose we have a given N-dimensional feature space $X = \{F_1, F_2, \ldots, F_N\}$. Let $C = \{P_1, P_2, ..., P_n\}$ be our data set of images. Then, any image P (for convenience we use P in place of P_i for any image from our collection) represented in this feature space can be writen in its histogram form $P = (p_1, p_2, \ldots, p_N)$ (see Section 2.1). Let a query image Q be represented in the same feature space X, $Q = (q_1, q_2, \ldots, q_N)$. What we seek now is a linear retrieval form which matches image queries against the collection images. A characteristic of the color space is that its colors are not independent, but correlated. Therefore, a retrieval form, when applied to images, should incorporate these correlations between the features. Further, a retrieval funtion which does not consider the correlations between features may give wrong ranking results.

We propose the following retrieval function, which includes a color-color similarity matrix A:

$$F : \mathbb{R}^{\mathbb{N}} \times \mathbb{R}^{\mathbb{N}} \rightarrow \mathbb{R}, F(P,Q) = P^t A Q, \tag{3}$$

where $A = (a_{ij})_{i,j=1,\ldots,N}$, is a symmetric similarity matrix with a_{ij} being the similarity between colors i, j (see Section 2.2). We assume $a_{ii} = 1$ and $0 \leq a_{ij} < 1$ for $i \neq j$. For example, matrix A can be computed as in the *QBIC* system (see Section 2.3).

By using the associativity property of matrix multiplication on the proposed retrieval function used to compute image-image similarity based on color, we obtain three color-color similarity based retrieval models:

i) Model *(PA, Q)*:

Transform all image histograms P by $P^{**} = A^t P$ and leave the query histogram, Q, unchanged. Then

$$F(P,Q) = (P^{**})^t \, Q \, .$$

ii) Model *(P, QA)*:

Transform the query histogram Q by $Q^{**} = AQ$ and leave the image histogram, P, unchanged. Then

$$F(P,Q) = P^t Q^{**} \, .$$

iii) Model *(PB, QB)*:

DEFINITION 1. A matrix A is positive definite iff $X^t A X > 0$, for all $X \neq 0,\ X \in \mathbb{R}^N$.

THEOREM 1. (Cholesky Factorization) If A is a $N \times N$ positive definite matrix, then there exists a unique $N \times N$ matrix B such that $A = B^t B$.

Now transform $P^* = BP$ and $Q^* = BQ$. Then

$$F(P,Q) \quad = \quad P^t A Q = P^t B^t B Q = (BP)^t (BQ) = (P^*)^t \, Q^* .$$

Notice that in order for the last model, *(PB, QB)*, to exists, a positive definite matrix A is required, otherwise the *Cholesky* decomposition cannot be applied (Golub and Loan, 1996).

Model *(PB, QB)* looks most appealing. Although $P^*$and $Q^*$are no longer histograms of the initial feature space. It employs a retrieval function symmetric in $P^*$and Q^*, and the retrieval function is a dot product that can be interpreted as a similarity. Also, both are vectors in the same new space, unlike in the other two models where the new forms (P^{**} and Q^{**}) are not any more in the same space with their associate vectors (Q and P, respectively).

By precomputing the similarity matrix and all the products *PA, PB, QA* and *QB,* the retrieval function becomes linear during the retrieval step. In the next section, we propose a more general form to compute the similarity matrix than the form used by the *QBIC* system.

3.2 Obtaining Similarity Matrix A

Let us suppose we have the conditions described in Section 2.2. We start with the assumption that there exists a proximity/distance function between any two colors. Then, we can easily transform proximities into similarities by using the following Lemma:

LEMMA 1. Let d be a proximity function on $A \times A$ and let $f : R^+ \rightarrow R^+$ be a strictly monotonic decreasing function. Then

$$s(x,y) = f(d(x,y))$$

for all $x, y \in A$, defines a similarity function.

Proof. We define $s_0 = f(d_0)$. Next, we check the axioms for a similarity function:

i) $d(x,y) \geq d(x,x) = d_0$. This implies $s(x,y) = f(d(x,y)) \leq f(d(x,x)) = f(d_0) = s(x,x) = s_0$.

ii) $s(x,y) = f(d(x,y)) = f(d(y,x)) = s(y,x)$.

iii) $s(x,y) = s_0$ implies $f(d(x,y)) = f(d_0)$ and $d(x,y) = d_0 \Rightarrow x = y$.

□

There are several ways to do this, some of them are given in (Smith, 1997; Hafner et al., 1995).
The corresponding similarities between any two features F_i and F_j are given as:

$$s(F_i, F_j) = a_{ij} = f(d_{ij})$$

(see Lemma 1). Thus, the similarity matrix can be considerated as having the following form:

$$A^* = \begin{pmatrix} 1 & \dots & s_0 f(d_{ij}) \\ \vdots & \ddots & \vdots \\ s_0 f(d_{ji}) & \dots & 1 \end{pmatrix},$$

where $s_0 > 0$ is a constant. Matrix A^* is symmetric ($d_{ij} = d_{ji}$, same distance) and $A^* = (A^*)^t$. Therefore, in order for matrix A^* to be positive definite, it must be diagonal dominant. This happens iff $0 \leq f(d_{ij}) < 1$ and some carefully chosen s_0. Matrix A^* represents a more general form of a similarity matrix and thus, it can be used as matrix A in our proposed retrieval form. From now on we use matrix A^*, but we refer to it as the similarity matrix A, for convenience.

If matrix A is not given, then from the distances between the features F_i and F_j we can derive their similarity, as:

$$a_{ij} = s(F_i, F_j) = \begin{cases} 1 & i = j, \\ s_0(1 - \frac{d_{ij}}{M}) & i \neq j, \end{cases} \tag{4}$$

where $M = \max\{d_{ij}, i \neq j\}$ and $0 < s_0 \leq 1$. In this case, if we choose s_0 such that $s_0(N-1)(1 - \frac{\min d_{ij}}{\max d_{ij}}) \leq 1$, then matrix A is positive definite (Zhao, 2000) (e.g. $s_0 = 1$). Under these conditions matrix A from Equation (4) is a positive definite symmetric similarity matrix. This implies the existence of a unique non-singular matrix B such that $A = B \cdot B^T$ (Theorem 1).

Note that the above matrix A represents a more general form of a similarity matrix than the *QBIC* similarity matrix. In particular, for $s_0 = 1$, it has the form of the *QBIC* similarity matrix.

We notice that the size of matrix A depends on the number of features used and not on the number of images from collection. Since we create the feature space during the image processing step, we can compute the similarity matrix during this step too, and just use it during retrieval. Next section establishes some characteristics of our similarity retrieval function.

3.3 Correctness of the Similarity Retrieval Approach

For any images P, Q equation

$$< P, Q >= P^t A Q$$

defines an inner product.

We want to show that our proposed retrieval form is a similarity function. First we describe some of its properties.

In order to show the effect of F, let us consider the case where Q consists of exactly one feature k, $Q = (q_i), q_k = 1, q_{i \neq k} = 0$. Then, for $P = (p_1, \ldots, p_N)$ we get

$$F(P, Q) = \sum_{j=1}^{N} a_{kj} p_j, \tag{5}$$

which shows that the contribution of each feature j in an image P depends on its frequency in P and its similarity to feature k.

We normalize all images P from collection with a factor $w = \sqrt{P^t A P} > 0$.

THEOREM 2. For any single color images P_i, P_j, containing colors F_i and F_j, respectively, the following holds:

i) $< P_i, P_j >$ is a strictly monotonic transformation of a_{ij}.

ii) $< P_i, P_i >= 1$.

iii) $\delta(P_i, P_j)$ is a strictly monotonic transformation of d_{ij}.

Proof.

i) For clarity, in here we use A^* and A with their original meaning. $< P_i, P_j >= P_i^t A^* P_j = P_i^t a_{ij}^* P_j = s_0 f(d_{ij}) = s_0 f(d(F_i, F_j)) = s_0 s(F_i, F_j) = s_0 a_{ij}$ (Equation (5)).

ii) $< P_i, P_i >= P_i^t A P_i = P_i^t a_{ii} P_i = P_i^t P_i = 1$.

iii) $\delta(P_i, P_j) = (P_i - P_j)^t A (P_i - P_j) = P_i^t A P_i - P_i^t A P_j - P_j^t A P_i + P_j^t A P_j = 2 - 2 P_i^t A P_j = 2 - 2 s_0 f(d_{ij})$. Because f is strictly monotonic decreasing, $\delta(P_i, P_j)$ is a strictly monotonic increasing function of d_{ij} .

□

THEOREM 3. The proposed retrieval function is a similarity function as defined in Lemma 1 with $s_0 = 1$.

Proof. Easy to prove, using *(i)* from Theorem 2 and the normalization factor. □

COROLLARY 1. $\delta(P, Q)$ is a strictly monotonic transformation of d_{ij} for any images P and Q, if P and Q are single-color images.

Proof. $\delta(P, Q) = (P - Q)^t A (P - Q) = 2 P^t A P - 2 P^t A Q = 2 - 2 P^t A Q = 2 - 2 s_0 f^*(d_{ij})$, where $f^*$is a strictly monotonic decreasing function $f^*(d_{ij}) = \sum_{i,j} f(d_{ij})$. □

COROLLARY 2. In the conditions of the Theorem 2, $F(P, Q) =< P, Q >= a_{ij}$ where P has only color i and Q has only color j, (see i) of the same theorem), which shows that the proposed form does not have the *distance mismatch* problem of the *QBIC* system, discussed in Section 2.4 .

Observation. If $A \equiv I$ (identity matrix) then

$$\delta(P, Q) = (P - Q)^T (P - Q)$$
$$and$$
$$< P, Q >= P^T Q$$

The last function is nothing but the inner product function used for text retrieval in Information Retrieval and represents a measure of similarity between the two document vectors (Raghavan and Wong, 1986). We will refer to this function as the *(P, Q)* model or the standard model. In the next section, we point out some characteristics of the above introduced retrieval models.

3.4 Discussion of the Retrieval Models

In this section we discuss some characteristics of the retrieval models.

The query image Q contains the features desired by user. The linear function F (Equation (3)) tries to match these desired features of the given query against each image from collection. Thus, the bigger the value of the F function applied to a query Q and an image P, the better the match between the query image Q and the collection image P, or, in other words, the closer the two images.

Notice that a linear form for the retrieval function is preferred due to the availability of several feedback methods. By precomputing the similarity matrix and all the products *PA*, *PB*, *QA* and *QB*, the retrieval function becomes linear during the retrieval step.

Image histograms are large feature vectors, which means many histogram bins are involved in computations during retrieval, resulting in a time consuming retrieval step. In order to improve the retrieval time we have to reduce the number of computations during this process. For this, we can pre-compute certain reusable information for all possible *(image, query)* pairs[1],[2] and then store these values in a database apriori, during a pre-processing step, and then, we can just read them during the retrieval step. This results in an optimization of number of computations during the retrieval process.

The standard retrieval function (model *(P, Q)*) has a linear time complexity with respect to the number of features. The first among the proposed similarity models that is, *(PA, Q)*, has also a linear time complexity during retrieval. Models *(P, AQ)* and *(PB, QB)* have a quadratic complexity with respect to the number of features if used as is, but by using the above pre-processing step, the complexity during retrieval becomes linear.

Note that for retrieval purposes we can use any of the above models. The main disadvantage of the standard retrieval model *(P, Q)* is that it does not consider the main characteristic of the color feature space: its elements are not independent vectors. That is, the features used to represent both image and queries are correlated. It is expected for these correlations to influence the retrieval process and therefore, they must be incorporated in the retrieval function. Thus, it could be predicted that the *(P, Q)* model will not give good results when used for a similarity search. To overcome this problem, we proposed the retrieval similarity models introduced in Section 3.1.

In Section 4.1 we conclude that all models are equivalent (including the standard one) when learning based on relevance feedback is employed, in the sense that the query learned in one model can also be learned in any of the other models, but if matrix A is known then the last similarity model *(PB, QB)* should be preferred. If matrix A is unknown, we believe that the system needs maybe more iterations during retrieval (more time to learn matrix A), but eventually it is able to learnt it. In some cases we do not even need to know matrix A (see next Example). That is, the standard model *(P, Q)* gives good results. In the following section we give a simplified example for our retrieval approach.

3.5 Example: Using the Similarity Retrieval Model

Suppose that we have the following feature space $X = (x_1, x_2, x_3, x_4)$, where $x_i = (r_i, g_i, b_i)$ are *RGB* colors given in Table 1 a). Suppose also that we have a collection of 5 images $P = \{P_1, P_2, P_3, P_4, P_5\}$ represented in the feature space X, as given in Table 1 b).

The set of colors and images were chosen such that they cover a wide variety of cases which may occur in an image collection, such that single color images (P_1, P_2, P_3, P_4) and multi-color image (P_5), pure colors (x_1, x_2, x_3 - red, green, blue - occuring in image $P_1, P_2,$ and $P_3,$ respectively) and a composite color (x_4 - purple - in image P_4).

[1] We consider that our query image is an image from the collection.

[2] The *(image, query)* pairs will be computed according to the chosen model and thus, they may no longer have the same initial meaning.

color bin	value
x_1	$(10, 0, 0)$
x_2	$(0, 10, 0)$
x_3	$(0, 0, 10)$
x_4	$(10, 0, 10)$

	P_1	P_2	P_3	P_4	P_5
x_1	8	0	0	0	4
x_2	0	8	0	0	4
x_3	0	0	8	0	0
x_4	0	0	0	8	0

Table 1: (a) *RGB* color bins; (b) Image Histograms.

The goal of this example is to analyze the behavior of the above described models for retrieval, without any kind of user subjectivity. For this, we simplify it as much as possible by keeping only the minimum information required to understand the correlations between colors within images and their influence on the retrieval.

From a user point of view, the above images are classified according to their feature(s) as: P_1 red image, P_2 green image, P_3 blue image, P_4 purple image, P_5 image with red and green in the same amount. What we want from our system is to get the same "feeling" of the images.

Logically, from the image feature description, for each image considered as a query we should expect an order of the images from the collection, as follows:

Query	Ordering
P_1	P_1, P_5, P_4, P_2, P_3
P_2	P_2, P_5, P_1, P_4, P_3
P_3	P_3, P_4, P_1, P_5, P_2
P_4	$P_4, \{P_3, P_1\}, P_5, P_2$
P_5	$P_5, \{P_1, P_2\}, P_4, P_3$

Table 2: Intuitive order.

where P_i is the query image followed by its ordered list of the collection data, with the most similar image first; the braces $\{.\}$ contain the list of items without any specific order (the same similarity to the query image).

In the sequel, we analyze the system behavior only in two cases, for the standard retrieval model *(P, Q)* and for the new model *(PB, QB)*. The justification is that both models keep their new transformed vectors in the same space (Section 3.1). First, we apply the normalization (described in Section 3.3) to each image. The normalization factor depends on the chosen model, *(P, Q)* or *(PB, QB)*, and it ensures that our proposed retrieval function is a similarity function. We record the resulting lists from our system for both retrieval models *(P, Q)* and *(PB, QB)*, in Table 3.

	(P, Q)	(PB, QB)
P_1	$P_1, P_5, \{P_2, P_3, P_4\}$	$P_1, P_5, P_4, \{P_2, P_3\}$
P_2	$P_2, P_5, \{P_1, P_3, P_4\}$	$P_2, P_5, \{P_1, P_3\}, P_4$
P_3	$P_3, \{P_1, P_2, P_4, P_5\}$	$P_3, P_4, P_5, \{P_1, P_2\}$
P_4	$P_4, \{P_1, P_2, P_3, P_5\}$	$P_4, \{P_1, P_3\}, P_5, P_2$
P_5	$P_5, \{P_1, P_2\}, \{P_4, P_3\}$	$P_5, \{P_1, P_2\}, P_4, P_3$

Table 3: Example results for *(P, Q)* and *(PB, QB)* models.

As a general observation of the results, the *(PB, QB)* model gives less images with the same similarity, or in other words, it features a better separation of ranks. We notice that the standard model *(P, Q)* misplaces the rankings for all queries, whereas the similarity model does not misplace in two cases (purple image P_4, and multi-color image P_5). Therefore, we conclude that the similarity model *(PB, QB)* deals well with composite color image and with the multi-color image, but it cannot handle the single color images completely. Although it does not give a perfect ranking, the proposed similarity model *(PB, QB)* seems to perform better than the standard model *(P, Q)*. As we can see from the example, there are still some more problems, for both models, to be solved in order to improve the results.

We know that the Euclidian distance used to compute the distance/similarity between colors does not give a correct order according to color hues. For example, it might happen for a red color to be closer to a green color than to a purple color. We believe that this way of computing the color-color distances is the real cause of our incorrect results in the above example. However, we do not deal here with this distance computation problem. One might find better way to classify colors than the distance d (Equation (1)) we used. We believe that a better color distance will give the desired orders for the proposed model. We conclude that a model which consider the similarity between colors (like our *(PB, QB)* model) might give better rankings than the standard model *(P, Q)*. Anyway, as we can see also from this example, the standard model might be sufficient in some cases (eg. query image P_2). Therefore, the choice of the model depends on the image dataset. The following section deals with the learning process, more precisely, with learning the user's "ideal" query.

4 Learning the Query

In this section, we prove that the above retrieval models are equivalent. Then, we introduce a new algorithm, *Kernel Rocchio Algorithm*, seeking to improve the retrieval process.

4.1 Learning by Using the New Models

As was already mentioned in the introduction, one of the advantages of a linear retrieval function is that there are several training methods of feedback available, such as Rocchio (Rocchio, 1971), Perceptron (Wong et al., 1988) or Support Vector Machine (Vapnik, 1995). The question arises which of the three retrieval models of Section 3 is most appropriate for learning. In this section we prove that from a theoretical viewpoint, the proposed models are equivalent, more exactly, that the query learned in one model can also be learned in any of the other models.

The three retrieval models that we considered in Section 3 differ in their image representation. If P is the original database image histogram, then in *(PA, Q)* model an image is represented as $P^{**} = A^t P$, in *(PB, QB)* model it is represented as $P^* = BP$, and in *(P, QA)* model P is not modified at all. If we do learning, after feedback from user, we cannot expect that the query Q is a color histogram, nor can we expect that it is a transformation of it such as $Q^* = BQ$, or $Q^{**} = Q^t A$, because each feedback from user modifies the initial query vector (independent of its initial form) to some other vector. The question now arises whether the different image representations make a difference for learning. To this end, we prove the following theorem:

THEOREM 4. Let A be a positive definite matrix. Then, for any query for one image representation, there exists a query for any other image representation (of the above described models) that produces exactly the same ranking.

Proof.

i) Let us assume that images are represented by their original color histogram P and let Q be any query. We want to show that there exists a corresponding query Q^* under the condition that a database image is

represented by $P^* = BP$ that gives the same ranking. Because A is positive definite, we know that B^{-1} exists. Let

$$Q^* = (B^{-1})^t Q\,.$$

Then,

$$(P^*)^t\, Q^* \quad = \quad (BP)^t((B^{-1})^t Q) = P^t B^t (B^{-1})^t Q == P^t \left(B^{-1}B\right)^t Q = P^t Q.$$

ii) Let us assume that images are represented as $P^{**} = AP$ and let Q be any query. We want to show that there exists a corresponding query Q^{**} under the condition that a database image is represented by its original histogram P that gives the same ranking. Let

$$Q^{**} = AQ.$$

Then

$$(P^{**})^t\, Q = (AP)^t Q = P^t AQ = P^t Q^{**}.$$

iii) Let us assume that images are represented as $P^* = BP$ and let Q^* be any query. We want to show that there exists a corresponding query Q^{**} under the condition that a database image is represented by $P^{**} = AP$ that gives exactly the same ranking. Because A is positive definite, we know that B^{-1} exists. Let

$$Q^{**} = B^{-1}Q^*.$$

Then,

$$(P^{**})^t\, Q^{**} = (AP)^t(B^{-1}Q^*) = P^t AB^{-1}Q^* = P^t B^t BB^{-1}Q^* = P^t B^t Q^* = (BP)^t Q^* = (P^*)^t Q^*.$$

□

As a result we see that whenever the images are linearly separable in one representation, they are also linearly separable in the other two representations.

The question now is which one is the best representation for color retrieval. If we assume that the color-color similarity matrix $A = B^t B$ is known, then $P^* = BP$ should be preferred because color retrieval can be done here either by using the query as a color histogram Q, which can be transformed into $Q^* = BQ$, or by using learning. On the other hand, if A is not known then, as far as the learning process by using a linear retrieval function is concerned, there is no major disadvantage. In the worst case, it could happen that the retrieval result of the initial query is rather poor and more iterations are necessary for learning. The algorithm that we propose for retrieval is reffered as *Kernel Rocchio Algorithm* and it is presented next.

4.2 Kernel Rocchio Algorithm

Let C = collection of images be our data set, $C = \{P_1, P_2, \ldots, P_n\}$, R = the set of the relevant images from the collection and $\bar{R}$ = the set of the non-relevant images. Then, $C = R \cup \bar{R}$ and $R \cap \bar{R} = \emptyset$.

The query learned by using the Rocchio method (Rocchio, 1971) is given by:

$$Q = \frac{1}{|R|} \sum_{P_i \in R} \frac{P_i}{\|P_i\|} - \frac{1}{|\bar{R}|} \sum_{P_j \in \bar{R}} \frac{P_j}{\|P_j\|},$$

where $\|.\|$ defines norm-2.

It can be proven that the RSV (retrieval status value) of any image P_k from the collection can be computed as:

$$RSV(\phi(P_k)) \quad = \quad \frac{1}{|R|} \sum_{P_i \in R} < \phi(P_i), \phi(P_k) > - \frac{1}{|\bar{R}|} \sum_{P_j \in \bar{R}} < \phi(P_j), \phi(P_k) >, \tag{6}$$

where $\phi : X^N \rightarrow \mathbb{R}^N$ is a mapping function from the N- dimensional feature space X to the real space, called the *feature map*. The greater the *RSV* of an image, the closer the image to the query *Q*. The mapping function ϕ is continuous and satisfies:

$$< \phi(P_i), \phi(P_k) >= K(P_i, P_k), \tag{7}$$

where $K : X \times X \rightarrow \mathbb{R}$ is a *kernel function* (Cucker and Smale, 2001; Schölkopf et al., 1999).

In the following, we give the kernel functions we used in our system:

i) Polynomial Product: $K(P, Q) = (< P, Q >)^p , p > 0$. In our implementation, we used $p = 1$ refered as the scalar product kernel, and $p = 2$ refered as the polynomial product kernel.

ii) Radial Basis: $K(P, Q) = \exp(-\frac{\|P-Q\|^2}{2\lambda^2}), \lambda \in \mathbb{R}^+$.

By using the *Rocchio Algorithm* (Equation (6)) with the kernel function (Equation (7)), we obtain the *Kernel Rocchio Algorithm* to compute the *RSV*s of the collection images, as follows:

$$RSV(\phi(P_k)) \quad = \quad \frac{1}{|R|} \sum_{P_i \in R} \frac{K(P_i, P_k)}{\sqrt{K(P_i, P_i) \cdot K(P_k, P_k)}} - \frac{1}{|\bar{R}|} \sum_{P_j \in \bar{R}} \frac{K(P_j, P_k)}{\sqrt{K(P_j, P_j) \cdot K(P_k, P_k)}}, \tag{8}$$

for any image P_k from the collection. The *Kernel Rocchio Method* combines the simplicity of Rocchio method with the power of non-linear kernel functions to improve the retrieval process.

Rocchio Algorithm is commonly used for retrieval purposes because of its simplicity and its power to quickly discriminate between the relevant and non-relevant data. But, like some other training methods, its major drawback is that it can only be used for linear classifiers. To overcome this, one can make use of non-linear kernel functions. The idea is to use this simple learning method, *Rocchio*, and the non-linear decision functions (such as kernels) in order to obtain faster results with the same computational cost. For this, we propose a new algorithm, the *Kernel Rocchio Method* described above.

The usage of the kernel functions can enhance the capabilities of a system, by providing more general decision surfaces in the decision space. By replacing the inner product $< \phi(P_i), \phi(P_k) >$ with a suitable kernel K, everything that has been done in the linear case can also be applied to the nonlinear case (Cucker and Smale, 2001; Schölkopf et al., 1999). Thus, the kernel functions can be used to replace the linear retrieval form required by some training methods of feedback (Rocchio, Perceptron, and Support Vector Machine).

5 Tests and Results

We have constructed an image retrieval system based on the algorithms developed in Section 3.1 and 4.2. For learning purpose, we used *Kernel Rocchio Method* with three different kernels, for all similarity retrieval models, including the standard model *(P, Q)*. We have evaluated the performance of the retrieval system by using the *R-Norm* (normalized recall) measure (Bollmann and Raghavan, 1998). We tested the above algorithms on the

same synthetic set of 100 images. The data set was built as follow: we choose 10 images of different sizes which are, as much as possible, very different from the color point of view. Then, each of these images was modified by using rotation, shifting, and translation transformations creating 9 new images. That is, there are 10 heterogenous images in the dataset, each one representing a subset of 10 homogeneous images. Finally, the whole dataset set was randomized. For each query image from the dataset its corresponding subset of homogeneous images constitutes the relevant set of images. In this way, we eliminate the user subjectivity during the testing process.

To test the system we used the *Test and Control Testing Method*. We randomly divided the image collection into two equivalent groups: a test set and a control set of images. The control set was used for feedback, the test set provided untested data for evaluation. We used the *pbmquant* Unix package which uses the median cut algorithm to quantize the dataset to the same set of 256 colors. The similarity matrix was computed by using Equation (4), based on the Euclidian distance between colors (Equation (1)).

Figure 1 compares the scalar kernel, polynomial kernel and radial basis kernel curves in the case of the standard retrieval model and the new retrieval model *(PB, QB)*.

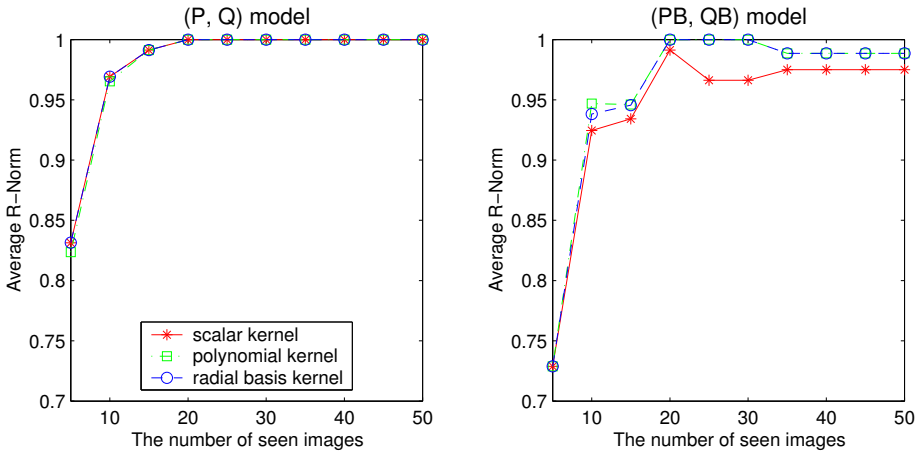

Figure 1: Kernel curves.

As we can see from the Figure 1, the two models seem to have the same kind of behavior, but the standard model seems to perform slightly better than the *(PB, QB)* retrieval model where the kernel curves decay after reaching the maximum value. In both models, all curves, except the one corresponding to the scalar kernel, reach a maximum value of 1 after 20 images. We notice that for the standard model all three curves depict the same behavior, with no difference between their values. For the similarity model, it seems that all theree curves are oscillating very close to each other. There is almost no difference between polynomial and radial basis kernels, whereas the scalar kernel displays a lower performance.

A problem of the new retrieval model *(PB, QB)* is the computation of the similarity matrix based on the Euclidian distance between colors. It is known that this distance does not discriminate well the colors. Therefore, we believe that a better measure of the distance between colors will improve the performance of the model.

Anyway, as we stated before the similarity matrix depends on the dataset, more precisely on the colors, and reflects their correlations. The way of computing the similarity matrix influence the similarity retrieval model *(PB, QB)*, and thus, it needs more attention in future research. We had built this dataset such that it is easy to create user judgement. In reality, this is not the case. For more complex image collections the system

might have a different behavior. Therefore, we consider that more experiments need to be done to analyze the proposed model. Next section concludes the paper.

6 Conclusion

In this paper, we address the problem of linear color retrieval in image database. Many applications require that the retrieval function reflects the dependencies between features, e.g. the correlations between colors. Whereas the standard retrieval form used in Information Retrieval for document retrieval ignores these correlations, our proposed similarity-based retrieval form fulfill this requirement leading to three new similarity retrieval models. Out of these models, the last one requires the similarity matrix to be positive definite. As a result, we proved that these models are equivalent, in the sense that the query learned in one model can also be learned in any of the other models.

By using the technique described in Section 3.4, the similarity retrieval function becomes linear during the retrieval step. A linear retrieval form allows us to use the learning algorithms already existing in the Information Retrieval literature. To compute the similarity matrix, we propose a more general form than that of the *QBIC* approach. In Section 4.2, we introduce a new algorithm, *Kernel Rocchio* algorithm, which combines the simplicity of Rocchio method with the power of non-linear kernel functions to improve the retrieval process.

We implemented our algorithms and tested them on a synthetic dataset which allowed easy user judgement of the image relevance. Our results show that the *(P, Q)* retrieval model used together with the polynomial kernel provides better results compared to other combinations of retrieval models and kernel functions. We think the reason for this unexpected result is that the way of computing the similarity matrix based on the Euclidian distance between colors is inappropriate. If the method of computing color correlations is improved, the similarity retrieval model should perform just as well as the *(P, Q)* model, as shown by our theoretical results.

In the future work, we plan to investigate better techniques for computing the similarity matrix and to perform more experiments on sufficiently large number of different images to analyze the effect of the similarity matrix on the retrieval process.

References

Alberto Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, Inc., 2001.

P. Bollmann and V. V. Raghavan. On the necessity of term dependence in a query space for weighted retrieval. *Journal of the American Society for Information Science*, (49):1161–1168, 1998.

Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39(1):1–49, 2001.

M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Janker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):310–315, 1995.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, chapter 4, pages 142–145. The Johns Hopkins University Press, 1996.

James Hafner, Harpreet S. Sawhney, Will Wquitz, Myron Flickner, and Wayne Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

Vijay V. Raghavan and S. K. M. Wong. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5):279–287, 1986.

J. J. Rocchio. *Relevance feedback in Information Retrieval*, chapter 14, pages 313–323. Prentice-Hall, Inc., 1971.

Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors. *Advances in Kernel Methods*. The MIT Press, Cambridge, Massachusetts, 1999.

Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.

John R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, analysis and Compression*. PhD thesis, Columbia University, 1997.

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

Remco C. Veltkamp and Mirela Tanase. Content-based image retrieval systems: A survey. Technical report, Ultrecht University, http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/, 2000.

S. K. M. Wong, Y. Yao, and P. Bollmann. Linear structure in information retrieval. In *Proceedings of the 11th SIGIR Conference*, pages 219–232, Grenoble, 1988.

Xiaoquan Zhao. Space transformation and clustering methods for proximity data set. Master's thesis, University of Louisiana at Lafayette, 2000.