# Analysis of Relevance Feedback in Content Based Image Retrieval

P. Suman Karthik
Center for Visual Information Technology
International Institute of Information Technology
Gachibowli, Hyderabad-500019, India
Email: sumankarthik@students.iiit.ac.in

C. V. Jawahar
Center for Visual Information Technology
International Institute of Information Technology
Gachibowli, Hyderabad-500019, India
Email: jawahar@iiit.ac.in

*Abstract: Relevance feedback in Content Based Image Retrieval(CBIR) has been an active field of research for quite some time now. Many schemes and techniques of relevance feedback exist with many assumptions and operating criteria. Yet there exist few ways of quantitatively measuring and comparing different relevance feedback algorithms. Such analysis is necessary if a CBIR system is to perform consistently. In this paper we propose an abstract model of a CBIR system where the effects of different modules over the entire system is observed. Using this model we thoroughly analyse performance a set of basic relevance feedback algorithms. Besides using standard measures like precision and recall we also suggest two new measures to gauge the performance of any contemporary CBIR system.*

## I. INTRODUCTION

Relevance feedback was introduced in Content Based Image Retrieval (CBIR) to improve the performance by human intervention[1], [2]. Since then it has become an integral part of most CBIR systems. There are a plethora of relevance feedback algorithms available in literature. Though there have been some studies on relevance feedback algorithms[3], [4], [5], there has been no systematic evaluation of the performance (stability, convergance, precision etc.) available. Such an analysis is both pertinent and necessary as there is a trend towards consistency in CBIR systems even in the face of a highly dynamic environment[6], [7]. Garunteed performance and stability of the system can be achieved only when all the factors internal and external that effect the system are identified, gauged and tracked. Their behaviour at all conditions shuld also be observed. In this paper, we attempt this with the help of a CBIR framework. which can be considered as a generalisation of many practical algorithms. We would like to make it explicit that our model does not cover the class of long-term learning and feature-less semantic indexing (eg. LSI) schemes. We analyze popular class of relevance feedback algorithms using an instance of the abstract model for its convergence, performance in presence of strong or week concepts in the collection of images. We also define and measure the performance of the algorithms using two new measures. Before explaining the performance analysis, we explain the experimental setting and the model we used for the analysis.

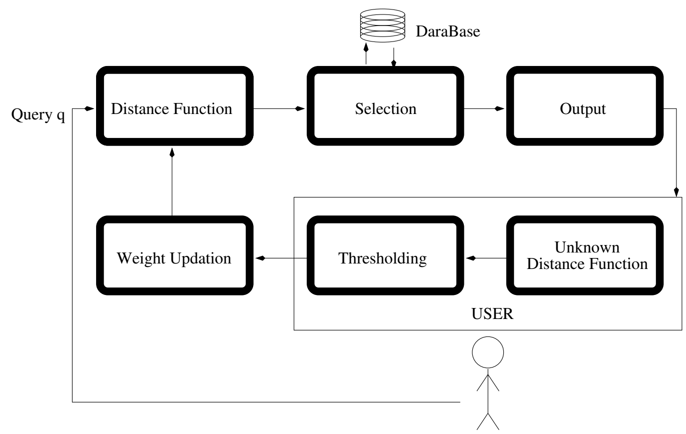


Fig. 1. Block Diagram of an Abstract CBIR Model

*Query and Database:* An ideal CBIR system can accept queries in many forms. We consider a query to be a sample image or an image patch, represented as a feature vector $q$ of dimension $L$. A typical image database $\mathcal{I}$ used for CBIR problems, has large number of images. Many of real-life collections will have strong concepts (classes/themes) present. In such cases, these datasets can be modeled as a union of clusters, possibly with outliers. In an arbitrary collection of images, it could be modeled as a set of random points in a feature space. Objective of CBIR systems is to identify the most similar images to the query image. Similarity is measured in terms of (weighted combination of) one or more features.

*Number of Results Displayed* The system at any query retrieves $M$ most relevant images based on the type of distance function. The user then indicates which images are relevant and which are not based on his concept (significance of features and distance function). Usually $M$ is a small number (say 10) and is independent of the database size or the number of acceptable images ($N$) present in the database. The total number of acceptable images in the database is unknown to the user. These are all the images in the dataset that fall within an acceptable margin of the user's concept.

*User Model* User accepts an image if it is within an acceptable distance from the query point. User may give a feedback of $\{1,0\}$, $\{1,0,-1\}$, [0,1], depending on the accept, reject status or the partial acceptabilities user give. In the simplest form user can be modeled as a weighted distance function, with weights represented as $W_u$. However, it may be noted that realistic models need not be metric or as simple as this. User compares the query image with the images returned by the system and accepts based on a *threshold of relevance* $\theta$. Comparison is done using a distance function $G(q, I_i, w_u)$, which returns the dissimilarity between the two images he is comparing.

*System Model* CBIR system learns to approximate the user notion with the help of relevance feedback. System also models the similarity with a distance function $F(q, I_i, w_s)$; however with a different (unknown) weight vector. During the learning process, system learns to approximate the similarity. While using a weighted distance function, system weight represents the relative importance of the particular feature for a given query. The change in system weights is the means by which the system tries to emulate the user's concept.

*Weight Updation Scheme* The user gives feedback to the images that are displayed by the system and the system has to improve from the feedback the user is giving. Where the system assumes that those features which are more common among the relevant examples might be more significant in being able to represent the concept. Detailed description of the relevant feedback techniques is provided in III.

### A. Performance Measures for the Analysis

The system is initialized with some rudimentary concept (system weights or system parameters) when the system retrieves the images, the user gives feedback about the discrepancy between his and the system's concept. Then the system re-estimates its concept based on the feedback to represent as closely as possible the user's concept. This concept updation is done by changing the system weights. The point at which the system's concept and the user's concept are fairly similar and consistently stays so is called the point of convergence. How fast this convergence takes place signifies how fast the system adapts to a user. In essence the difference in concept, which is the difference between weights, reduces until convergence. Hence the difference of weights at each iteration forms a good metric for measuring the speed and efficiency of the system.

The number of relevant images retrieved can also be used as a performance measure. This is an intersection of the set containing the $M$ images and the set containing $N$ images which are all the relevant images in the database.

Precision and recall are also popular for characterizing the performance. The total number of relevant images is calculated by comparing the query image with all the images in the database and the user assigning score to each of them, and finally thresholding the scores to get the total number of relevant images to the concept the user is looking. The number of relevant images is calculated by cross checking with the relevant images in the database. The retrieved set is obtained by nearest neighbor search.

Rate Of Convergence(ROC), not to be confused with receiver operator characteristics is one of the measures that we propose to evaluate the performance of a CBIR system. ROC is the number of iterations after which the precision of the system remains constant or the system parameters do not change considerably. This measure is very apt in the contemporary scheme of things as users of a realistic CBIR system expect fast and accurate results with the least amount of iterations.

Rate Of Ascent(ROA) is the second measure we propose. It is the measure that quantifies the performance of an algorithm. It is

$$\frac{2 * precision * recall}{N * (precision + recall)}$$

Where $N$ is the Rate of Convergence of the relevance feedback module. The above measure simultaneously shows us the performance of the system in terms of precision, recall and efficiency of the system.

## II. Experimental Setting for the Empirical Analysis

For the implementation of any relevance feedback system the following basic assumptions are necessary. The features of the system can distinguish between relevant and irrelevant images. Relevant images are a small part of a large database. The relevant images can be clustered based on the features in the system. User gives accurate feedback.

### A. Database

Three kinds of databases were considered for the experimental study. A database without clusters(strong concepts), a database with clusters accompanied by random query points, and a database with clusters and query points that led in and around the clusters.

### B. Algorithm

Once the user submits a query to the system the images in the database are ranked based on the system distance function.The $M$ least dissimilar(most similar) images are selected as the relevant set. Then the user module compares the selected images with the query using the user similarity function and gives the $M$ dissimilarity values to the system. The dissimilarity values are classified into relevant and irrelevant

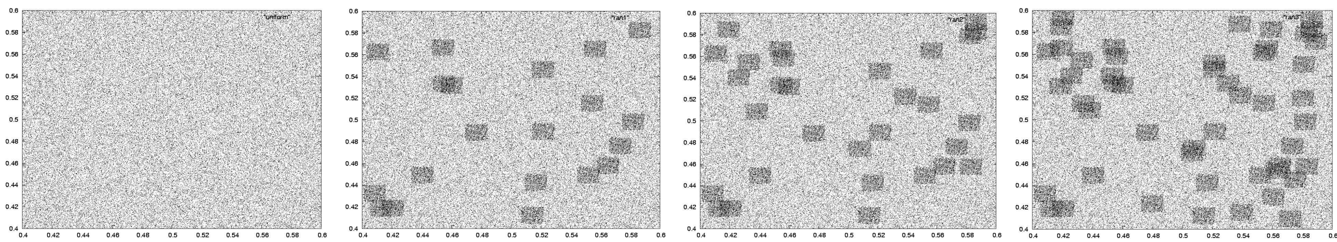

Fig. 2. The above figure shows two synthetic data sets D1,D2,D3,D4 . D1 is a sparse data set of 100000 points where the points in the feature space are spread uniformly with no clustering, indicating many independent concepts. D2 is a Desnse data set of 120000 points, D3 of 130000 points and D4 of 150000 points, where beside the uniform background distribution of D1 there are strong clusters of points present at different places in the feature space.

images using the threshold $T$. The relevance data is taken and relevant changes are made. The system continues again from the selection with these new weights.

## III. Relevance Feedback Techniques

The traditional relevance feedback framework is more or less the same in all systems. Initially a query is given to the system and a set of images retrieved. The user comments on or indicates which images in the set are, relevant and irrelevant. The system then takes the user's suggestion and tries to refine the retrieval scheme to achieve optimal retrieval performance. It is in this refinement and selection that various CBIR systems differ from each other.

### A. Statistical Methods

These were the earliest methods of heuristic weight adjustments. They used the nature of the distribution of relevant data in the features space to effectively cluster relevant examples. Most of these methods try to take advantage of the fact that under certain transformations the image database can be clustered into relevant and irrelevant images Or where the relevant images become clustered and the irrelevant ones become sparsely dispersed. The relevance feedback data is used to achieve this transformation. The Delta Mean algorithm for instance tries to find what features can effectively discriminate between the set of relevant samples($n_p$) and the set of irrelevant($n_n$) examples. This is done by calculating the Importance of each feature as the difference of the means of $n_p$ and $n_n$ images over that feature normalized by the sum of their standard deviations[8]. This is a simple algorithm that guarantees that it will give greater importance to the features that effectively separate both negative and positive examples. This algorithm has certain drawbacks. It assumes that the distribution of both the relevant and irrelevant images in the database are unimodal, but more often than not the $n_n$ images tend to violate this assumption. Another of its flaws is that it is sensitive to sample set size, because a small sample cannot successfully estimate the true standard deviation of the complete set. Most of these problems arise because this method treats the CBIR as a strongly constrained 2-class problem instead of a weakly constrained multi-class problem. Inverse variance and inverse sigma methods are better over the delta mean because they are much more weakly constrained. These methods take advantage of the fact that the ability of a feature to cluster the relevant images is inversely proportional to the variance and standard deviation of the relevant image set over that particular feature. These methods too have certain drawbacks the main cause of which is again the assumption of a unimodal distribution of the $n_p$ images in the feature space. They also fail to take advantage of the $n_n$ images. The membership criterion method[8] makes use of the $n_n$ samples with the $n_p$ ones without making any assumption about the nature of distribution of $n_n$. At the same time it still imposes a unimodal constraint on $n_p$. Here the mean and standard deviation of relevant set is used to calculate a hypothesis of importance of a particular feature and then this value is cross checked seeing what members of $n_n$ and $n_p$ fall into the relevant cluster. It is more or less a trail and error based algorithm where it tries out various constrained hypothesis to arrive at the one that most closely resembles the user model with respect to the choice of relevant and irrelevant images. The major flaw in this method is that $n_p$ is assumed to be unimodal which is rarely the case in the real world. This is because the user interprets the images with higher level features that have some remnants in the lower level features of the CBIR system but do not exactly map to the lower level in an ideal way and hence creating multi-modal distributions. Query Point Movement(QPM) and Query Expansion are two other methods that try to find an ideal query point from which the best possible and the highest $n_p$ can be achieved. The variants of these methods can make use of both $n_p$ and $n_n$ to arrive at this new query point. In QPM one simply finds the centroid of $n_p$ which acts as the new query point. In query expansion instead of assuming a unimodal distribution the system assumes many smaller unimodal distributions to construct multiple centroids using QPM on individual clusters of relevant samples and then the multiple centroids are taken as multi-point query and images are retrieved from iso-similarity regions based on these points. The main disadvantage of QPM is the constraint of unimodality on $n_p$ and inability to make effective use of $n_n$ data when its not unimodal.

| Rate Of Convergence | | | | |
|---|---|---|---|---|
| Algorithm | D1 | D2 | D3 | D4 |
| Inverse Sigma | 3 | 6 | 8 | 11 |
| Delta Mean | 15 | 15 | 15 | 15 |
| MC($\theta = 1$) | 2 | 3 | 2 | 2 |
| QPM | 15 | 15 | 15 | 15 |
| KLDivergence | 15 | 15 | 15 | 15 |
| Parzen | 15 | 11 | 9 | 3 |
| BDA | 4 | 6 | 9 | 11 |
| SVM | 7 | 5 | 7 | 6 |

TABLE I

NO OF ITERATIONS AFTER WHICH PRECISION REMAINS CONSTANT FOR D1,D2,D3 AND D4

*1) Comments:*

*a) Performance:* The performance of the algorithms is quantified in the form of precision, recall, Rate Of Convergence and Rate Of Ascent. We see that Inverse Sigma outperforms, delta mean, QPM(Rocchio) and membership criterion($\theta = 1$) When $n_p$ is unimodal. We can also observe that statistical method membership criterion out performs the other 2 though it only manages to beat InverseSigma by a small margin. Both delta mean and QPM suffer from a high Rate Of Convergence and are apparently unstable. This can be seen in Table 1 and Table 2.

*b) Absence of Strong Concept:* When there was an absence of strong concepts in the database the relative precision in the database seemed to be low and hence the number of relevant images retrieved. Even when strong concepts were present the precision was not better if the query points did not belong to the concepts. Once the queries were close to or belonged to a concept the precision shot up while the recall plummeted as a result the ability of the system to generalize to a concept suffered.

*c) Complex User Models:* When faced with user models that are not unimodal the performance of the statistical methods drops considerably. The algorithms were run on different distance functions other than the Euclidean and was seen that the algorithms performed well as long as the data was unimodal in nature. That is the reason why all the algorithms showed good or better performance when the user distance function was Minkowski or Manhattan.

### B. Kernel Based Methods

These methods use some kind of kernels to achieve relevance feedback. In Parzen Window Based Density Estimation[9] the authors use Bayesian inference to classify the images as relevant or irrelevant. In order to do this one requires the knowledge of the densities $p(X|n_p)$ and $p(X|n_n)$. These densities can be estimated using parametric or non-parametric methods. Here the non-parametric method is preferred to the parametric one as the parametric methods impose unimodal constraints over the distribution of the data. The non-parametric method used is a Parzen window method with a Gaussian kernel that acts as a smoothing function. Here all the features are assumed to be independent for real-time performance of the system. Once the densities are estimated one can go ahead with Bayesian inference of the database with the above densities. And with the relevance feedback of this step the whole process starts over again for the next step. Most of the recent work on relevance feedback has been concentrated on SVMs[10] or Support Vector Machines. SVMs are used to classify linearly inseparable classes by using a reproducing kernel. This is done by first projecting all the data points onto a higher dimension where they are linearly separable, there they are classified. The objective of an SVM is to find a hyperplane Such that the distance from the plane to the closest of point in $n_p$ and $n_n$ is maximized. A detailed explanation of the possible kernels and SVMs is beyond the scope of this document. One class SVMs have also been used to estimate density of the positive and negative distributions. There are many advantages to using SVMs for relevance feedback.

- No significant constraints are placed on the target, like unimodality.
- The kernel can be tuned to perform well for static applications
- They are less sensitive than density based methods to imbalance between positive and negative sample because they only use support vectors. However they are sensitive to small sample sizes.

All the above make SVMs good for relevance feedback. In BDA and KBDA[11], a CBIR system is treated as a one positive and many negative clas problem as explained in [11]. This means that while the positive class is clustered the negative class can be scattered all over the feature space. BDA is about finding the linear transformation that has the most scatter of negative images over the scatter of the positive images. In kBDA the transformation is converted into its inner-product form to account for the non-linear nature of the data.

*1) Comments:*

*a) Performance:* The kernel based methods show a varied range of performance. Since here the relevant and irrelevant examples are linearly separable the choice of kernels doesn't play a big part in the performance of the algorithms. We see from the performance table that BDA out performs the other two in both rate of convergence and rate of ascent there by proving to be clearly a good choice. But on the other hand the Parzen window based density approximation algorithm has the unique advantage of being a method that is progressive with every retrieval. The performance can be seen in Table1 and Table2.

| | Precision | | | | Recall | | | | ROA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | D1 | D2 | D3 | D4 | D1 | D2 | D3 | D4 | D1 | D2 | D3 | D4 |
| Inverse Sigma | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Delta Mean | 68 | 71 | 74 | 76 | 70 | 73 | 74 | 76 | 14 | 26 | 31 | 56 |
| MC($\theta = 1$) | 88 | 89 | 73 | 62 | 92 | 94 | 71 | 61 | 135 | 211 | 278 | 301 |
| QPM | 89 | 72 | 67 | 58 | 95 | 78 | 69 | 56 | 18 | 21 | 27 | 29 |
| KLDivergence | 93 | 89 | 77 | 70 | 95 | 89 | 78 | 70 | 19 | 31 | 47 | 51 |
| Parzen | 86 | 97 | 109 | 128 | 89 | 98 | 114 | 128 | 18 | 56 | 352 | 432 |
| BDA | 99 | 94 | 85 | 86 | 99 | 93 | 84 | 87 | 74 | 78 | 77 | 98 |
| SVM | 92 | 92 | 87 | 93 | 94 | 94 | 89 | 95 | 40 | 63 | 212 | 310 |

TABLE II

PERCENTAGE OF PRECISION, RECALL AND RATE OF ASCENT TO VALUES OF INVERSE SIGMA FOR D1, D2, D3 AND D4

*b) Absence of Strong Concept:* These are effected in the same ways as the other classes of algorithms. This is because the user selection function remains the same across all the algorithms and this is solely influences the precision and recall in a major way.

*c) Complex User Models:* These algorithms were designed to adapt to complex user models. Here Parzen window based density does not use kernels for projection into a higher dimensional space where the classes are linearly separable like the other two algorithms do. This means that SVMs and kBDA are better suited to deal with complex user models. In our experiment though the complex user models divide the feature space into linearly separable classes. Hence the performance of kernels need not be broached.

### C. Entropy Based Methods

ntropy is an estimation of the deviation of a random variable from pure randomness. In weight adjustment based on entropy of $n_p$, the entropy of all the features for $n_p$ is estimated. The expectation is that if a feature has the ability to cluster the positive examples then its entropy will be low. Entropy is very attractive because no assumptions or constraints need be made on the distribution of the data. A variant of this method takes advantage of $n_n$ provided by the user along with the nature of the $n_p$ distribution. Here one predict that the best feature is one that gives a non random distribution for $n_p$ as well as $n_n$[8].Here there is ambiguity in the sense that even features that can't discriminate $n_p$ and $n_n$ well will achieve a high score. KL Distance or Divergence forms a sort of dissimilarity measure based on entropy. It is based on the cross entropy between the two distributions and the entropy of the main distribution. The KL Distance does not follow the triangle law of inequality. KL Distance between two distributions can be different based on whose entropy is being calculated. The main problem with direct KL-Divergence is the apparent lack of symmetry. A variant of this method makes KL-divergence much more sound by taking into account KL-divergence from the $n_n$ too along with KL-divergence from $n_p$. This theoretically forms a great measure for relevance and discriminative power because of its apparent lack of constraints on the distribution of the data.

#### 1) Comments:

*a) Performance:* The performance of the entropy based algorithms matched or in most cases bettered the performance of the conventional statistical algorithms. Though speculatively small sample set was considered a challenge for this class of algorithms, experimentally it held its own against the other algorithms by returning approximately the same or better number of relevant images by the fifth iteration. The performance of the algorithms under unimodal circumstances can be seen in Table1 and Table2.

*b) Absence of Strong Concept:* When there was an absence of strong concepts in the database the relative precision in the database seemed to be low and hence the number of relevant images retrieved. Even when strong concepts were present the precision was not better if the query points did not belong to the concepts. But once the queries were close to or belonged to a concept the Precision shot up while the recall plummeted as a result the ability of the system to generalize to a concept suffered.

*c) Complex User Models:* Even in the complex models the entropy based algorithms equaled or bettered the performance of the other algorithms. When faced with user models that are not unimodal the performance of these methods also drops considerably. The algorithms were run on different distance functions other than the Euclidean and was seen that the algorithms performed well as long as the condition of unimodality in the target data was met. That is the reason why all the algorithms showed good or better performance when the user distance function was Minkowski or Manhattan. The problem here is not with the entropy based methods of weight updation but with the selection schema that is based on unimodal criteria.

### D. Other Schemes

By no means is the above list of methods exhaustive on the ways of applying relevance feedback. There have been many other methods and are bound to be many more. Some of the other prominent ones are SOMs(Self Organizing Maps) in which using the $n_p$ and $n_n$ maps are constructed that have the ability to place the positive and negative impulses on different areas of the map. For new feedback a better or a new map is built based on $n_p$ and $n_n$. There are also other methods ranging from Decision Trees to Bayesian Estimation[12] of the user behavior. The methods listed above are just a few of the plethora of the relevance feedback algorithms out there.

## IV. Discussion

From the above one can see that the statistical models perform well as relevance feedback modules and so do some other algorithms. The results above were the consequence of a very primitive user model. This model may not only be flawed in its replication of the realworld user models but may also be unable to replicate the complexity of its real world counter part. Yet in the absence of large amounts of data providing real world interaction between user and a CBIR system any attempt to replicate the model would be futile and hence a simple tractable synthetic model should suffice for these studies for now. The same goes for the sparse and dense data sets that are taken here in the experiment. Another important factor that needs a mention here is one of the major factors affecting the stability of the relevance feedback algorithm for any given data set at any given point. This is the difference in the density of points in the feature space around the query. If the result set is M and the relevant set is N then it has been observed that the greater the difference between M and N the more the relevance feedback system tends to become unstable. This is because at a lower M vs N difference the user is able to check any fluctuations in the learning of the concept by providing adequate feedback. On the other hand A high M vs N difference would either lead to the relevance feedback module learning a Specialisation or Generalisation of the concept the user is searching for. This behaviour can be clearly seen in Figure 3 where one can see the difference in user and system models on y axis and the difference between M and N on the x axis, as one can see the best learning takes place when the difference between M and N is zero. The above mentioned factors are but a few of the many that govern the performance of relevance feedback systems in the real world. The ability to track and observe all of them will only be possible by building much more complex models that are learnt from realworld systems and their user interactions.

## V. Conclusion

The choice of different parameters and algorithms effects a general CBIR system in profound ways. The behavior of the system under all circumstances cannot be predicted. Seldom is a CBIR system fine tuned and optimized for its role in image retrieval. This is because of the plenty of configurations a system can exist and perform and the difficulty in pin pointing the aims of a CBIR system.

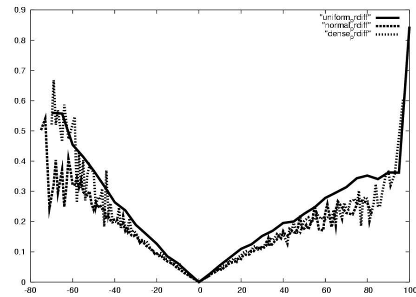


Fig. 3. The above figure plots the difference between the user and system model on the y-axis and the difference between M and N on the x-axis. One can clearly observe that the best learning of user model takes place when M is Equal to N

The Present work hopes to throw some light on the above issues and opens the door for flexible CBIR systems that can be tuned at runtime ensuring that the system runs at its optimal performance in the current stage or nature of the system. We have also suggested two performance measures that are useful for the quantitative and qualitative analysis of any CBIR system with relevance feedback.

## References


[1] T. S. H. Yong Rui and S.-F. Chang, "Image retrieval: Past, present, and future," in *International Symposium on Multimedia Information Processing*, 1997.

[2] S. Aksoy, R. Haralick, F. Cheikh, and M. Gabbouj, "A weighted distance approach to relevance feedback," in *International Conference on Pattern Recongnition*, 2000, pp. Vol IV: 812–815.

[3] X. Zhou and T. Huang, "Exploring the nature and variants of relevance feedback," in *CBAIVL01*, 2001, pp. 94–100.

[4] A. Doulamis and N. Doulamis, "Performance evaluation of euclidean/correlation-based relevance feedback algorithms in content-based image retrieval systems," in *International Conference on Image Processing*, 2003, pp. I: 737–740.

[5] T. Huang and X. Zhou, "Image retrieval with relevance feedback: From heuristic weight adjustment to optimal learning methods," in *International Conference on Image Processing*, 2001, pp. III: 2–5.

[6] B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T. Huang, "Visualization and user-modeling for browsing personal photo libraries," *International Journal of Computer Vision*, vol. 56, no. 1-2, pp. 109–130, January 2004.

[7] J. Yang, Q. Li, and Y. Zhuang, "Towards data-adaptive and user-adaptive image retrieval by peer indexing," *International Journal of Computer Vision*, vol. 56, no. 1-2, pp. 47–63, January 2004.

[8] A. F. V. Shiv Naga Prasad and S. Rakshit, "Feature selection in example-based image retrieval systems," in *International Conference on Vision, Graphics and Image Processing*, 2002.

[9] C. Meilhac and C. Nastar, "Relevance feedback and category search in image databases," in *International Conference on Multimedia Communications Systems, Vol. 1*, 1999, pp. 512–517.

[10] P. Hong, Q. Tian, and T. Huang, "Incorporate support vector machines to content-based image retrieval with relevant feedback," in *Pengyu Hong, Qi Tian, Thomas S. Huang. Incorporate Support Vector Machines to Content-based Image Retrieval with Relevant Feedback. Image Processing, 2000. Proceedings. pp. 750-753.*, 2000.

[11] X. S. Z. Thomas S. Huang, "Image retrieval with relevance feedback: From heuristic weight adjustment to optimal learning methods," in *International Conference on Image Processing*, 2001, pp. 2–5.

[12] I. Cox, M. Miller, T. Minka, T. Papathornas, and P. Yianilos, "The bayesian image retrieval system, pichunter: Theory, implementation,and psychophysical experiments," in *Tran. On Image Processing, Volume 9, Issue 1, pp. 20-37, Jan. 2000.*, 2000.