Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

*IJCSMC, Vol. 2, Issue. 5, May 2013, pg.155 – 161*

**RESEARCH ARTICLE**

# Content Based Image Retrieval with Log Based Relevance Feedback Using Combination of Query Expansion and Query Point movement

**Bhailal Limbasiya[1], Swati Patel[2]**

[1]Department of Computer Science & Technology, Gujarat Technological University, Ahmedabad, India
[2]Assistance professor, L. D. College of Engineering, Gujarat Technological University, Ahmedabad, India

*[1] bhailal.ldce@gmail.com; [2] swati.ldce@gmail.com*

***Abstract— This paper presents the Log based relevance feedback techniques, which combines two popular techniques of relevance feedback: query point movement and query expansion. From the past experiments, these two techniques are giving good results for image retrieval. But query point movement is limited by a constraint of unimodality in taking into account the user feedbacks. Query expansion gives better results than query point movement, but it cannot take into account irrelevant images from the user feedbacks. We combine the two techniques to profit from their advantages and to cope with their limitations. From a single point initial query, query expansion provides a multiple point query, which is then enhanced using query point movement. To learn the multiple point queries, the irrelevant feedback images are classified into query points which are clustered from relevant images using the query expansion technique. The experiments show that our method gives better results in comparison with the two techniques of relevance feedback taken individually.***

***Key Terms: - Relevance feedback; Query expansion; Query point movement; Log-based relevance feedback***

## I. INTRODUCTION

Content based image retrieval (CBIR) has received much attention in the last decade, which is motivated by the need to efficiently handle the rapidly growing amount of multimedia data. Content based image retrieval is the technologies that retrieve images from a very large data base by their low level visual features such as color, texture and shape. It covers versatile areas, such as image segmentation, image feature extraction, representation, mapping of features to semantics, storage and indexing, image similarity-distance measurement and retrieval making CBIR system development a challenging task.

Many CBIR systems have been developed, including QBIC [1], Photobook [2], MARS [3], NeTra [4], PicHunter [5], Blobworld [6], VisualSEEK [2], SIMPLIcity [7]. Many researchers in information-technology field and leading academic institutions try to develop content based image retrieval system for very large image database. Recently researcher focus in CBIR has moved to an interactive mechanism called Relevance feedback that involves a human as part of the retrieval process.[8],[4] In this approach, the retrieval process is interactive. To search for desirable images, a user provides the query image, and the system returns a set of similar images based on the extracted features. In CBIR systems with relevance feedback (RF), a user can mark returned images, which are then fed back into the systems as a new refined query for the next round of retrieval. Given the difficulty in learning the users' information needs from their feedback, multiple rounds of relevance feedback are usually required before satisfactory results are achieved. As a result, the relevance feedback phase can be extremely time-consuming. Moreover, the procedure of specifying the relevance of images in relevance feedback is usually viewed as a tedious and boring step by most users. Hence, it is required for a CBIR system with relevance feedback to achieve satisfactory results within a few feedback steps as possible, preferably in only one step. Despite previous efforts to accelerate relevance feedback using active learning techniques, traditional relevance feedback techniques are ineffective when the relevant samples are scarce in the initial retrieval results. From a long-term learning perspective, log data of accumulated users' relevance feedback could be used as an important resource to aid the relevance feedback task in CBIR. Although there have been a few studies carried out on the exploitation of users' log data in document retrieval, little research effort has been dedicated to the relevance feedback problem in CBIR. The paper is organized as follows: A Log based relevance feedback technique is described in Section II. In Section III, we describe the proposed work. Finally, we conclude the paper in Section IV.

## II. RELATED WORK

We first give an for log-based relevance feedback that systematically integrates the log data of users' relevance judgments with regular relevance feedback for image retrieval. Fig. 1 shows the architecture of the proposed system. First, a user launches a query in a CBIR system for searching desired images in databases. Then, the CBIR system computes the similarity between the user query and the image samples in database using the low-level image features. Images with high similarity measure are returned to the user. Next, the user judges the relevance of the initially returned results and submits his or her judgments to the CBIR system. A relevance feedback algorithm refines the initial retrieval results based on the user's relevance judgments, and returns an improved set of results to the user. Typically, a number of rounds of users' relevance feedback are needed to achieve satisfactory results.

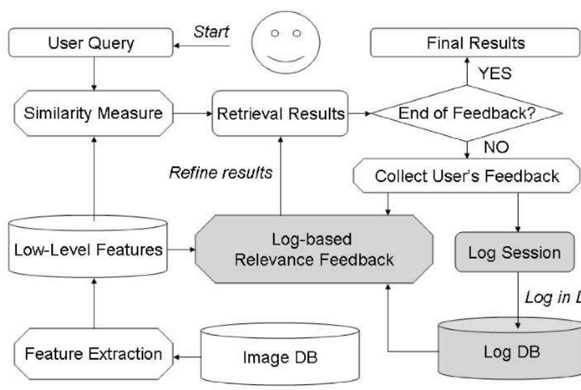



Fig. 1. The architecture of Log Based Relevance Feedback system

In Fig1, we see that the online relevance feedback from users is collected and stored in a log database. When feedback log data is unavailable, the log-based relevance feedback algorithm behaves exactly like a regular relevance feedback algorithm.

Log based Relevance Feedback Using Query Expansion

Query expansion has been shown to be effective in exploiting user query log data in traditional document information retrieval. We extend it to log-based relevance feedback [9] for image retrieval. Log-based relevance feedback with query expansion can be described as follows: When feedback log data is unavailable, the log-based relevance feedback algorithm behaves exactly like a regular relevance feedback algorithm, which learns the correlation between low-level features and users' information needs through the feedback image examples. So first consider that there is no log is available then it works like given below.

To support multiple query points [10], we extend the query model to include multiple points and a distance aggregation function. We denote the *n* query points by $p_1$, $p_2$,....., $p_n$ and the aggregate distance function for feature representation x by $A_{i,j}$ . The overall query model thus becomes < ($p_1$, $p_2$,....., $p_n$) , $d_{i,j,k}$ , $A_{i,j}$ >. Below we discuss our multipoint query technique based on query expansion. While the multipoint technique can work in conjunction with changes to the distance function $d_{i,j,k}$, changes to $d_{i,j,k}$ are orthogonal to the multipoint technique

The approach of Query Expansion is an aggregation function based on a weighted summation of the distances to the query points. Let there be a weight $w_t$ for each query point *pt* with $1 \le t \le n$, $0 \le w_t \le 1$, and $\sum_{t=1}^{n} w_t = 1$, that is, there is a weight between 0 and 1 for each query point, and all weights add up to 1. The distance aggregation function $A_{i,j}(x)$ for computing the distance of a feature representation value (i.e.,f an object) *x* to the query is $A_{i,j}(x) = \sum_{t=1}^{n} w_t \, d_{i,j,k}(x, p_t)$. Initially, all the weights are initialized to $w_t = 1/n$, that is, they are all equal. To compute a new query using relevance feedback, two things change: (i) the set of query points $p_t$ and (ii) the weights $w_t$ that correspond to each query point $p_t$.

Relevant points are added to the query if they are near images that the user marked as relevant. The rationale is that those images are good candidates for inclusion since they are similar to other relevant images and thus represent other relevant images. After the user marks all relevant images, the system computes the similarity between all pairs of relevant and query points in order to determine the right candidates to add. For example, suppose that an initial query contains two sample points $p_1$ and $p_2$. The system returns a ranked list of objects< $a_1$, $a_2$, ..., $a_m$>, since $p_1$ and $p_2$ were in the query, *a*1 = *p*1 and *a*2 = *p*2. The user marks *a*1, *a*2, and *a*4 as relevant and *a*3 as very relevant. Next, the system creates a distance table of relevant objects:

| *Point* | *rf* | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|---|
| $a_1$ | 1 (relevant) | 0 | 0.4 | 0.1 | 0.5 |
| $a_2$ | 1 (relevant) | 0.4 | 0 | 0.4 | 0.4 |
| $a_3$ | 2 (very relevant) | 0.05 | 0.2 | 0 | 0.2 |
| $a_4$ | 1 (relevant) | 0.5 | 0.4 | 0.4 | 0 |
| $\sum$ | | 0.95 | 1 | 0.9 | 1.1 |

The value in column *r* row *s* in the table is computed as $d_{i,j,k}(a_r, a_s)/rfs$.1 Objects with a low distance are added to the query and the objects with a high distance are dropped from the query. In this example, *a*3 is added to the query since it has the lowest distance among points not in the query, and *a*2 is dropped from the query since it has the highest distance among the query points.

These query points are relevant to the user query so, it return to the user and this process is continue till user is not satisfy and then these results are store in to the log database.When the relevance feedback log is available at that time it work as given below. Assume a user labels N images in each round of regular relevance feedback, which is called a log session in this paper. Thus, each log session contains N evaluated images that are marked as either “relevant” or “irrelevant.” For the convenience of representation, we construct a relevance matrix (R) that includes the relevance judgements from all log sessions.

Fig. 2 shows an example of such a matrix. In this figure, we see that each column of a relevance matrix represents an image example in the image database, and each row represents a log session from the log database. When an image is judged as “relevant” in a log session, the corresponding cell in matrix R is assigned to the value þ1. Similarly, _1 is assigned when an image is judged as “irrelevant.” For images that are not judged in a log session, the corresponding cells in R are assigned to zero values. A user must first present a query q, either by providing a query image. Let Z = {$z_1, z_2, \ldots\ldots, z_{N_{img}}$} is the identity of images in the image database. Let X= {$x_1, x_2, \ldots\ldots, x_{N_{img}}$} is the image database, where each $X_i$ is a vector that contains the low-level features of the image $Z_i$. Let R= {$r_1, r_2, \ldots\ldots, r_{N_{log}}$} is the log data in the log database, where each $r_i$ contains relevance judgements in the ith log session. Let L = {($Z_1, y_1$),($Z_2, y_2$),…,($Z_{N_l}, y_{N_l}$)} be the collection of labeled images acquired through the online feedback for a user using Query Expansion.


Fig.2. The Relevance matrix for presenting the log information of user feedback

According to the technique, both the low-level features of the image content, i.e., X, and the log data of users' feedback, i.e., R, should be included to determine the relevance function $f_q$. The relevance function depends on both R and X, a simple strategy is to first learn a relevance function for each of these two types of information, and then combine them through a unified scheme.

$$f_q(Z_i) = \frac{1}{2}(f_R(Z_i) + f_X(Z_i))$$

Here, we will describe how to find out the relevance functions $f_R(Z_i)$ and $f_X(Z_i)$ separately. To estimate the similarity between two images $Z_i$ and $Z_j$, we suggest a modified correlation function to measure their relevance judgments in the log data, i.e.,

$$c_{i,j} = \sum_k \delta_{k,i,j} \cdot r_{k,i} \cdot r_{k,j},$$

Define as follows:

$$\delta_{k,i,j} = \begin{cases} 1 & \text{if} \quad r_{k,i} + r_{k,j} \geq 0, \\ 0 & \text{if} \quad r_{k,i} + r_{k,j} < 0. \end{cases}$$

Based on the above similarity function, we can develop the relevance function based on the log data. Let $L^+$ denote the set of positive (or relevant) images in L, and $L^-$ denote the set of negative (or irrelevant) samples. For an image in the database, we compute its overall similarities to both positive and negative images, and the difference between these two similarities will indicate the relevance of the image to the user's query. More specifically, the overall relevance function can be formulated as follows:

$$f_{\mathbf{R}}(\mathbf{z}_i) = \max_{k \in \mathcal{L}^+} \left\{ \frac{c_{k,i}}{\max_j c_{k,j}} \right\} - \max_{k \in \mathcal{L}^-} \left\{ \frac{c_{k,i}}{\max_j c_{k,j}} \right\}$$

After obtaining the relevance function on the log data, we can use it in learning the relevance function on the low level image features. Learning the relevance function on the image features is a standard relevance feedback problem in content-based image retrieval. For the low level features here we are using the, we measure their Euclidean distance $f_{EU}(Z_i, Z_j^+)$ based on the low-level image features. The final relevance score $f_Q(Z_i)$ for each image $Z_i$ is determined by the combination of $f_{EU}(Z_i, Z_j^+)$ and $f_R(Z_i)$, i.e. $f_Q(Z_i) = f_R(Z_i) + min_i f_{EU}(Z_i, Z_j^+)$. Images with the largest relevance scores will be returned to the users. As with the query expansion approach for standard relevance feedback, images that are already labelled as negative will be excluded from the retrieval list.

## III. PROPOSED WORK

A combination of query point movement and query expansion [11] is proposed to overcome problems related to query expansion and query point movement. The main drawback of query point movement is the unimodality on relevant examples that cannot be always verified. We solve this problem by using a clustering technique to build multiple local clusters that provide local unimodality using relevant examples. The main drawback of query expansion is the inability to make effective use of irrelevant examples. In our approach, we propose a sequential combination of the two techniques: first query expansion (Fig. 3b) then query point movement (Fig. 3c).

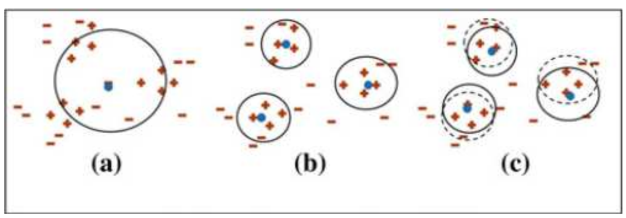



Fig. 3. Combination of Query Expansion and Query Point Movement

We are taking advantage of irrelevant examples using the technique of query point movement on multiple local clusters created using query expansion. We believe this sequential combination is the best among all possible combinations because it ensures the unimodality constraint and makes use of irrelevant examples (Fig. 3c) to effectively achieve the ideal query. The opposite combination (first query point movement then query expansion) is not good as query expansion cannot profit from irrelevant examples which were used in query point movement.

The purpose of this technique is to reach the ideal query through interaction with the user and to overcome the identified problems for both query point movement and query expansion. The first relevance feedback interaction loop is shown in Fig. 4. Initially, a single point query is formalized by using the feature vector of an image query q: Q = f1, f2, ..., fn fi is a n-dimension vector in the feature space. Then images are retrieved, the first N images are shown to the user. The user identifies and labels relevant/irrelevant images in an interaction process of RF, with the assumption that relevant examples in the result do not ensure the unimodality (Fig. 4, steps 1 and 2). Basing on (only) relevant/irrelevant images returned from the user the technique will replace and improve the single point query q by a multiple point query qi, query with multiple feature vectors using the two main processes: query expansion and query movement.

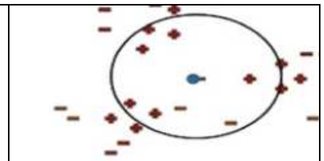
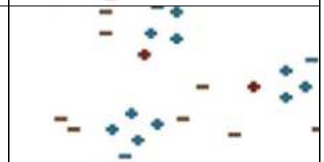
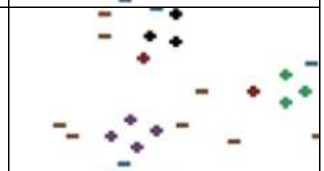
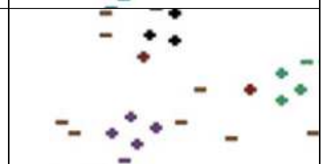
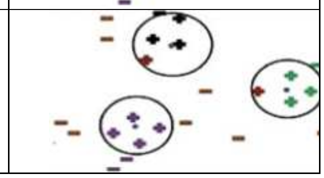
| | |
|---|---|
| (1)CBIR The blue dot is initial query and "+" is for relevant and "-" is for irrelevant. |  |
| (2)User Feedback "+" and "-" signs are marked by user during the relevance feedback |  |
| (3)Clustering of relevant image in multiple clusters. |  |
| (4)Classification of the irrelevant examples within the clusters of relevant examples. |  |
| (5)Construct the multiple point query using Rocchio's techniques. |  |

Fig. 4. Main steps of Combination of Query Expansion and Query Point Movement

First, the single point query q is expanded into a multiple point query to ensure the unimodality which is the problem of query point movement (Fig. 4, step 3): the relevant examples are clustered into c groups C1, C2, …, Cc. The number of clusters c is selected automatically using an adaptive clustering technique and is limited to a maximum value. In this step, we try to have the cluster/group maximums that are always unimodal. Clustering algorithms used in our system are presented in the end of this section. Second, in order to find the ideal points of the c relevant groups, the query point movement technique is used: irrelevant examples are classified into these c groups (Fig. 4, step 4) to identify irrelevant examples present in each local group. Relevant and irrelevant examples in each group are then used to build the multiple point query by the Eq. 1 (Fig. 4, step 5) in which we try to move the query points closer to the relevant images and away from the irrelevant images. The classifier k Nearest Neighbors (k-NN) is used in step 4 for the classification of irrelevant examples because of its efficiency and simplicity, the parameter k of the classifier is selected as follows:

k=min($|C_i|$, i=1:c)

and the query point $\overrightarrow{q_i}$ of cluster I is calculated using the Rocchio's formula:

$$\overrightarrow{q_i} = \frac{\sum_{j=1}^{m} \overrightarrow{R_j}}{m} - \frac{\sum_{j=1}^{n} \overrightarrow{I_j}}{n}$$

where I1, I2, …, In: n irrelevant examples and R1, R2, …, Rm : m relevant examples of the local cluster Ci. These c points of query form the final multiple point query. These multi point queries contains only relevant images and it is not contains irrelevant images.

Based on that we store these results in our database or we can call it user's log files. User's log files are very important to minimize the number of browsing. After that we know the procedure of log based relevance feedback which is describe above , we follow the procedure and get the most relevant images form the database and we display them to user and we doing this procedure until user satisfied.

**Algorithm:**

**Input:** A Query sample provided by the user q

**Output:** A set of images most relevant to the query q

1. Submit Query sample q to the retrieval system
2. If the log file available then go to step 11
3. If the results are unsatisfactory go to step 4
4. Otherwise record log file of feedback information of the current retrieval and stop

5. User submit feedback sample list
6. If this the first iteration

Single point Query
Compute the distance between the images and query q
Multi Point Query
Compute the distance between the images and each point of the query
Compute the final distance by combining all computed distance

7. Relevance feedback

if image marked as relevant than add in to the relevant set
if image marked as irrelevant than add in the irrelevant set

8. If this is the first interaction loop than cluster relevant images into c cluster
9. If this the first interaction loop

Classify images of the irrelevant set into c previous cluster
Else
Classify images of the relevant/irrelevant set into c previous cluster

10. Query Modification

Construct the multiple point Query using formula given below

$$Q' = \alpha Q + \beta\left(\frac{1}{N'_R}\sum_{i=1}^{N_R} F_i\right) - \gamma\left(\frac{1}{N'_N}\sum_{j=1}^{N_N} F_j\right)$$

Delete the relevant/irrelevant set
Go to step 2 for next iteration

11. Find the relevance function
12. Fine the low level feature function
13. Calculate the final relevance score and return the most relevant image to the user

Go to step 1 for next iteration.

## IV. CONCLUSION

Here, we are proposing a new method for relevance feedback. It is combination of two existing techniques of relevance feedback scheme: query point movement and query expansion. Taking advantage of irrelevant images and advantages of both traditional techniques, our method gives better results. By combining both techniques of query modification that are query point movement and query expansion, these two approaches can benefit from irrelevant examples. Our method does not require complex computations, but offers very significant improvements in accuracy compared to traditional techniques. As the relevance feedback methods presented here are valid for both text and image retrieval, we are planning, in the near future, to extend our cluster-based relevance feedback by combining together text-based and content based image retrieval. To achieve this, a text/image learning model is needed and can be built onto the same relevance feedback model. This learning model would be considered as long-term memory relevance feedback.

## REFERENCES

[1] M. Flikner, H. S. Sawhney, J. Ashley, Q. Huang, B. Dom, M. Gorkani, L. Hafuer, D. Lee, D. Petkovic, D. Steele and P. Yanker. “Query by image and video content: The QBIC system” Computer , vol.28, no.9, pp.23,32, Sep 1995.
[2] J. R. Smith and S.-F. Chang. VisualSEEk: “A fully automated content based image query system”. In Proceedings of the 4th ACM Multimedia Conference, pages 87-98,1996.
[3] M. Ortega-Binderberger and S. Mehrotra. “Relevance feedback techniques in the MARS image retrieval systems”. Multimedia Systems, 9(6):535-547, 2004.
[4] W. Y. Ma and B. Manjunath. Netra: “A toolbox for navigating large image databases”. In Proceedings of the iEEE international conference on image Processing, pages 568-571, 1997.
[5] J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. “The Bayesian image retrieval system”, PicHunter: theory, implementation, and psychophysical experiments. iEEE Transactions on image Processing, 9(1):20-37, 2000.
[6] Carson, C.; Belongie, S.; Greenspan, H.; Malik, J., "Blobworld: image segmentation using expectation-maximization and its application to image querying," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.8, pp.1026,1038, Aug 2002.
[7] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: “Semantics-sensitive Integrated matching for picture libraries”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(9):947–963, 2001.
[8] Thomas S.Huang , Yong Rui, “Image retrieval Past present and future" Proc of Int Symposium on

Multimedia information Processing ,Dec 1997
[9] Hoi, S.C.H.; Lyu, M.R.; Jin, R.; , "A unified log-based relevance feedback scheme for image retrieval," Knowledge and Data Engineering, IEEE Transactions on , vol.18, no.4, pp. 509-524, April 2006.
[10] Porkaew, K.; Ortega, M.; Mehrota, S., "Query reformulation for content based multimedia retrieval in MARS," Multimedia Computing and Systems, 1999. IEEE International Conference on , vol.2, no., pp.747,751 vol.2, Jul 1999
[11] Nhu-Van Nguyen; Boucher, A.; Ogier, J.; Tabbone, S., "Clusters-Based Relevance Feedback for CBIR: A Combination of Query Movement and Query Expansion," Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on , vol., no., pp.1,6, 1-4 Nov. 2010