0.0in

A Relevance Feedback Architecture for Content-based Multimedia Information Retrieval Systems *

Yong Rui, Thomas S. Huang   Sharad Mehrotra, and Michael Ortega

IFP Lab, Beckman Institute   Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Abstract

Content-based multimedia information retrieval (MIR) has become one of the most active research areas in the past few years. Many retrieval approaches based on extracting and representing visual properties of multimedia data have been developed. While these approaches establish the viability of MIR based on visual features, techniques for incorporating human expertise directly during the query process to improve retrieval performance have not drawn enough attention. To address this limitation, this paper introduces a Human-Computer Interaction based approach to MIR in which the user guides the system during retrieval using relevance feedback. Our experiments show that the retrieval performance improves significantly by incorporating humans in the retrieval process.

1 Introduction

While advances in technology allow us to generate, transmit, and store large amount of digital images and video, research in multimedia information retrieval (MIR) is still at its infancy. Most existing approaches to MIR belong to one of the following two categories. The first approach is based on annotating multimedia data with text and then using existing text information retrieval (TIR) engines (e.g., INQUERY [1], SMART [2]) to search for the visual information *indirectly* by using the annotations.

The other approach is to represent multimedia objects in the database using their visual features *directly* and can be summarized as follows:

1. Computer vision techniques are used to extract low level visual features from multimedia objects. For example, color, texture, shape features for images, and motion parameters for video.
2. For a given feature, a representation of the feature and a notion of similarity measure are determined. For example, color histogram is used to represent color feature, and intersection distance is used for similarity measure.
3. Objects are represented as a collection of features and retrieval of objects is performed based on computing similarity in the feature space. The results are ranked on the similarity values computed.

Due to the difficulty in capturing the content of multimedia objects using textual annotations and the non-scalability of the approach to large data sets (due to a high degree of manual effort required in defining the annotations), the approach based on supporting content-based retrieval over visual features has become a promising research direction. This is evidenced by several prototypes[3, 4, 5, 6] and commercial systems[7, 8] that have been built recently. While these existing systems successfully establish the viability of the approach, techniques for incorporating human expertise directly during the query process to improve retrieval performance have not drawn enough attention.

In the information retrieval literature it has been well established that retrieval performance can be significantly improved by incorporating the user as part of the retrieval loop[2, 9]. *Relevance feedback* is the mechanism supported by TIR systems to enable users to guide the computer's search for relevant documents. In the relevance feedback approach, the system returns to a user an initial set of answers which the user marks as being relevant or not relevant. Using the relevance feedback the system refines the initial query until the user is satisfied.

The relevance feedback technique has been used in multimedia database annotation[10]. During the annotation, the system learns by positive and nega-

*This work was supported in part by NSF/DARPA/NASA DLI Program under Cooperative Agreement 94-11318, in part by ARL Cooperative Agreement No. DAAL01-96-2-0003, and in part by Research Board and CSE, UIUC

tive examples provided by a database annotator or user. Based on the examples, the system tries to annotate similar image regions both within the image and across the images. For a database with 1008 images, several hundreds of positive and negative examples need to be fed to the system to obtain a reasonable performance[10]. While this process is valid in annotation, or database pre-processing before retrieval, it is not very suitable for *real-time* retrieval process. Furthermore, different users have different perceptions of visual features. One person (database annotator) 's annotation might not suit another person (database user) 's perception.

To make the MIR system truly adaptive to different users in *real-time*, this paper introduces a two-layer relevance feedback architecture which we have implemented in the Multimedia Analysis and Retrieval System (MARS)[6, 11, 12, 13, 14, 15, 16]. Our preliminary experiments show that the MIR performance can be improved considerably by using the proposed approach.

The rest of the paper is developed as follows. A brief review and discussion of TIR models and relevance feedback is given in section 2. In section 3 we discuss the MIR object model and describe how relevance feedback can be used for multimedia retrieval. The proposed two-layer feedback architecture for MIR is discussed in details in section 4. Experimental results and conclusions are in sections 5 and 6 respectively.

2 TIR models and Relevance Feedback

A TIR system consists of a document model, a query model, and a model for computing similarity between the documents and the queries. The specification of each of these defines a *retrieval model*. One of the most popular retrieval models is the vector model[17, 2, 9].

Define w_k to be the weight for a keyword (term) t_k in document D, $k = 1, ..., N$, where N is the number of keywords. In the vector model, a document D is represented as a keyword weight vector.

$$D = [w_1; ...; w_k; ...; w_N] \quad (1)$$

Normally the weights are estimated by the product of *term frequency (tf)* and *inverse document frequency (idf)*. The *tf* factor reflects how frequently a term appears in a document. The higher the frequency, the higher the weight. The *idf* factor reflects the frequency that a term appears in the document collection. If a term appears in many documents in the collection, it is not a good discriminator between documents. Hence it should be assigned a low weight. Experiments have shown that the product of *tf* and *idf* is a good estimation of the weights[17, 2, 9].

The query Q has the same model as that of document D. When a query is submitted, it is represented as a weight vector

$$Q = [w_{q1}; ...; w_{qk}; ...; w_{qN}] \quad (2)$$

The similarity between D and Q is defined as the Cosine distance.

$$Sim(D, Q) = \frac{D\ Q}{||D||\ ||Q||} \quad (3)$$

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

In the vector model, the specification of w_{qk}'s in Q is very critical, since the similarity values ($Sim(D, Q)$'s) are computed based on them. However, it is usually difficult for a user to express his information need precisely; thus w_{qk}'s may not be accurate. To overcome this impreciseness, the technique of *relevance feedback* is used[2, 9, 17]. Relevance feedback is the process of automatically adjusting an existing query using information fed-back by users about the relevance of previously retrieved documents.

The mechanism of this method can be described elegantly in the vector space. If the sets of relevant documents (D_R) and non-relevant documents (D_N) are known, the optimal query can be proven to be[17, 2, 9]

$$Q_{opt} = \frac{1}{N_R} \sum_{i \in D_R} D_i - \frac{1}{N_T - N_R} \sum_{i \in D_N} D_i \quad (4)$$

where N_R is the number of documents in D_R and N_T the number of the total documents.

In practice, D_R and D_N are not known in advance. However, the relevance feedback obtained from the user furnishes approximations to D_R and D_N, which are referred as, D'_R and D'_N.

The original query Q can be modified by putting more weights on the relevant terms and less weights on the non-relevant terms.

$$Q' = \alpha Q + \beta\left(\frac{1}{N_{R'}} \sum_{i \in D'_R} D_i\right) - \gamma\left(\frac{1}{N_{N'}} \sum_{i \in D'_N} D_i\right) \quad (5)$$

where α, β and γ are suitable constants[2, 9]; $N_{R'}$ and $N_{N'}$ are the numbers of documents in D'_R and D'_N. Q' approaches Q_{opt}, as the relevance feedback iteration moves on. Experiments show that the retrieval performance can be improved considerably by using the relevance feedback [17, 2, 9].

3 Multimedia Object Model in MIR

Before we describe how the relevance feedback technique can be used for MIR, we first need to formalize how a multimedia object is modeled. A multimedia object O_M is represented as:

$$O_M = O_M(D, F, R, M, V) \quad (6)$$

- D is the raw data of the object, e.g. a JPEG image, or an MPEG video.
- $F = \{f_i\}$ is the set of features associated with the object, e.g. color, texture, and shape for images; motion parameters for video.
- $R = \{r_j\}$ is the set of representations for a given feature f_i , e.g. both color histogram and color moments are representations for color feature.
- $M = \{m_k\}$ is the set of similarity measures. Some examples are Cosine, Euclidean, histogram intersection, etc. For a given feature f_i, m_k is combined with r_j to determine how f_i will be perceived, e.g. color histogram (r_j) and histogram intersection (m_k) together determine how the color feature (f_i) is perceived.
- $V = \{v_j\}$ is the set of realizations for set R. For each r_j there is a v_j which stores the actual values for that representation. That is, v_j is an instance (realization) for the corresponding r_j. (Note the same index j is used in both r and v.)

Since a query Q_M itself is a multimedia object, the model of O_M is also the model for the query object.

The proposed multimedia object model is illustrated in Figure 1. The top architecture is for general multimedia objects. The bottom architecture is an example of how the model is used to describe an image object.

Because of the rich content in the multimedia objects, the proposed model supports multi-element (F, R, M) sets.

- Multi-element F set
  The information contained in a multimedia object is too rich to be captured by a single feature alone. Various features, such as color, texture, shape, layout, motion parameters, etc, are extracted to make the MIR system flexible enough to support different information need of different users. For example, while one user may wish to retrieve an image based on its color feature, another user may wish to retrieve the same image based on texture feature.

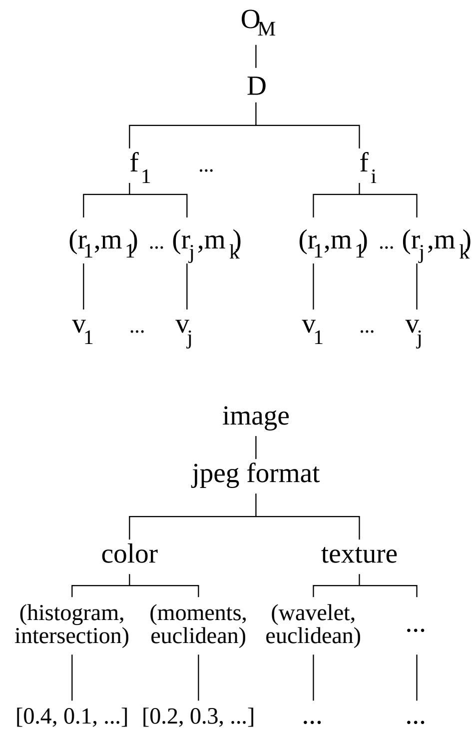

Figure 1: Multimedia object model

- Multi-element (R, M) set
  For any given f_i, there exist dozens of (r_j, m_k) combinations, none of which have been proven to be the best in simulating user's perception for that feature. We might never find such a (r_j, m_k) combination, since different persons, or even the same person at different circumstances, will have different perception criteria. Therefore, rather than prefixing a single-element (R, M) set at the system design stage, if (R, M) is implemented as a multi-element set, the MIR system would be more flexible to support different perception criteria of different users.

While the proposed model has the flexibility to support multi-element (F, R, M) sets, the MIR system must first identify which (f_i, r_j, m_k) combination best fits a particular user's information need, before the similarity $Sim(Q_M, O_M)$ is evaluated.

The query Q_M itself is a multimedia object and has the same model as that of the objects O_M's in the database. The selection of the best (f_i, r_j, m_k) can be done in either the query end or the object end. Considering there is only one query object Q_M while there may be thousands of objects O_M's in the database, the selection of the best (f_i, r_j, m_k) combination is always done in the *query end*. For selecting (f_i, r_j, m_k) at the query end, we have the following observations.

- At the F layer, it is easy for a user to specify which f_i's he is interested in. For example, the user can easily determine if he is interested in color or texture features of an object. This feature-specific query process is the basis of most existing MIR systems, as discussed in section 1. Different weights[8] and Boolean combinations[11] can also be attached to f_i's of interest, to form a combined query.

- While it is relatively easy for a user to specify the f_i's of interest, it is difficult for him to specify which (r_j, m_k) best matches his perception criterion, since this requires the user to have some knowledge in Computer Vision, which is normally not the case. This difficulty is bypassed by most existing systems by prefixing (r_j, m_k) at the system design stage at the cost of potentially poor retrieval performance. A technique of *automatic* (r_j, m_k) *selection* is needed to support MIR's flexibility for retrieval.

- At the V layer, it is even more difficult for the user to specify what are the exact values in v_j for his query Q_M. Little research has been done in this aspect to improve the query's accuracy of reflecting the user's information need. A technique of *query* v_j *refinement* is needed to allow the user to start the retrieval process with a coarse initial query.

Based on the above observations, a two-layer relevance feedback architecture is proposed to solve the difficulties of (r_j, m_k) *selection* and v_j *refinement* at (R, M) and V layers.

- *Top layer relevance feedback*: Automatic (r_j, m_k) selection
  For a given f_i that the user is interested in, the best (r_i, m_k) will be determined via relevance feedback. The user is not required to have any knowledge in Computer Vision, he only needs to rank the retrieval returns according to his own perception criterion and feedbacks the ranks to the computer. From the user's feedback, the computer will *automatically* identify the (r_j, m_k) that best fits this particular user's perception criterion.

- *Bottom layer relevance feedback*: Query v_j refinement
  Instead of specifying the exact values in v_j for his query, the user submits a coarse initial query to start the retrieval process. The values in v_j of the query will be *automatically* refined by the computer according to the user's feedback, such that the refined query is a better approximation to the user's information need . The query can be continuously refined until the user is satisfied or the refinement reaches the saturation point.

4 Integrating Relevance Feedbacks in MIR

This section discusses in details how the two-layer relevance feedback can be used for MIR to improve its performance. As mentioned in section 3, the relevance feedback technique will be performed in the query Q_M end.

4.1 Top Layer Relevance Feedback

To simplify the notations, define $P = (R, M)$. That is, P is the set of perception criteria which is determined by (R, M) combinations. Consequently, the selection of best (r_j, m_k) in (R, M) is equivalent to the selection of best p_t in P, where t is the index used for p.

For a given f_i, a set of useful p_t's are identified and represented in P. The procedure of automatic p_t selection is summarized as follows[14]:

1. The user specifies how many retrieval returns he wants to have. Let this number be N_r.

2. For an arbitrary given query, for each image I_n in the collection, $n = 1, ..., N_c$, where N_c is the number of images in the collection, compute the similarity distance $dist_{I_n,t}$ for each p_t in P.

3. For each p_t, based on $dist_{I_n,t}$'s, sort the image id's and construct a length-αN_r rank list l_t:
$$l_t = [I_{1,t}, ..., I_{m,t}, ..., I_{\alpha N_r,t}] \quad (7)$$
where α is a small positive integer greater than one, and $I_{m,t}$ is the image id for the mth most similar image to the query image when p_t is used. The reason we maintain a length-αN_r, not a length-N_r, rank list, is that these rank list l_t's are intermediate entities, a longer rank list will ensure better final precision. Experimentally we find that $\alpha = 2$ gives good final precision and has fast enough computation speed. Therefore, in the remaining of the procedure $\alpha = 2$ is used.

4. Define a rank-of operator $RANK_t(I_n)$, which finds the rank of image I_n, when p_t is used:
$$\begin{aligned} RANK_t(I_n) &= rank\ of\ I_n\ in\ l_t, & (8)\\ & if I_n \in l_t & (9)\\ RANK_t(I_n) &= 2N_r + 1, & (10)\\ & if I_n \notin l_t & (11)\end{aligned}$$

In Equations (8)-(11), for simplicity, we assign the same rank $2N_r + 1$ to all the images who are not in l_t.

5. For each image, compute the overall rank $rankAll_{I_n}$. Since only N_r images, where N_r is normally a small number, need to be returned to the user, there is no need to compute the overall rank for all the images in the database. To achieve fast retrieval speed, only the $rankAll_{I_n}$'s of the images appearing in some l_t's are computed. This approach results in a significant improvement in retrieval speed, while causing almost no retrieval miss.
$$rankAll_{I_n} = \sum_{t=1}^{T} RANK_t(I_n) \tag{12}$$
where T is the number of elements in P, and I_n appears in at least one of l_t's.
6. Based on $rankAll_{I_n}$'s, construct a length-N_r combined rank list l_c, which contains the overall most similar N_r images to the query image:
$$l_c = [I_{1,c}, ..., I_{m,c}, ..., I_{N_r,c}] \tag{13}$$
and send the retrieved image $I_{m,c}$'s to the user in the order specified in l_c;
7. The ranks for the retrieved images in l_c might not be the same as the user's perception and the user sends back a modified feedback rank list l_f:
$$l_f = [I_{1,f}, ..., I_{m,f}, ..., I_{N_r,f}] \tag{14}$$
8. For each l_t, compute the rank difference rd_t
$$rd_t = \sum_{m=1}^{N_r} abs(RANK_f(I_{m,f}) - RANK_t(I_{m,f})) \tag{15}$$
where abs denotes taking absolute value.
9. Return to the user the best p_{t^*}:
$$t^* = arg\ min(rd_t) \tag{16}$$
where arg denotes the index-selecting operator.

Usually this feedback procedure needs to be done only once and the subsequent retrieval is based on p_{t^*} just identified. Here, we assume a user's perception criterion stays relatively stable during the query process, which is normally a short period. If a user does find his perception is changing, a new round of feedback can be performed.

An alternative to the above standard procedure is to use multiple p_t's with different weights. Instead of selecting the best p_t with the minimum rank difference, we can use the inverse rank difference as the weight for each p_t. By incorporating multiple p_t's, although the retrieval speed is not as good as the above procedure, the retrieval precision is normally higher.

In both the standard and alternative relevance feedback procedure, the user is not required to have any knowledge of the characteristics of the perception criteria p_t's. He only needs to rank the retrieval returns according to his own judgment, and feedback the ranks to the VIR system. The good perception criteria p_t's will be automatically determined by the system based on the user's feedback.

4.2 Bottom Layer Relevance Feedback

The top layer's feedback provides the flexibility to support different perception criteria of different users; thus improving the MIR system's performance.

We can further improve the MIR performance by using the relevance feedback technique described in section 2 at the V layer[15].

The relevance feedback technique described in section 2 is a powerful technique applicable in the vector retrieval model. In the vector retrieval model, each document's content is captured by the weights of the keywords. The product of *tf* and *idf* is a good estimation of the weights[2, 9].

The counterpart of the weight vector $D = [w_1; ...; w_k; ...; w_N]$ in the vector model is v_j in the MIR model. Motivated by the two factors of *tf* and *idf*, similar factors are proposed in MIR to convert v_j to a weight vector w_j such that the relevance feedback technique described in section 2 can be applied to refine the query.

While the values in v_j represent the magnitude of the components, they do not have the physical meaning of *frequency*. Further more, the components in v_j may be defined over different physical domains. Their dynamic ranges vary drastically. For example, if we use the standard deviations in wavelet sub-bands as the texture feature representation, the values of standard deviations vary drastically across different wavelet sub-bands.

Motivated by *tf*, a factor of *component importance (ci)* is proposed and defined as[15]
$$ci_j = [\frac{v_{j1}}{mean_{j1}}, ..., \frac{v_{jq}}{mean_{jq}}, ..., \frac{v_{jN_v}}{mean_{jN_v}}]$$
where $mean_{jq}$ is the mean of the q^{th} component in v_j over all the objects in the collection, and N_v the length of vector v_j. Now, the components in ci_j are defined over the same domain, the same as the components in D in the vector model.

Motivated by *idf*, a factor of *inverse collection importance (ici)* is proposed and defined as[15]
$$ici_j = [log_2(\sigma_{j1}+2), ..., log_2(\sigma_{jq}+2), ..., log_2(\sigma_{jN_v}+2)]$$

where σ_{jq} is the standard deviation of the q^{th} component in ci_j over all the objects in the collection. Note that, ici penalizes those components who have small discriminating power but favors those having large discriminating power. This intuition justifies the standard deviation σ being a good measure of ici.

The weight vector w_j is obtained as

$$w_j = ci_j \times ici_j \tag{17}$$

After the conversion from v_j to w_j, the relevance feedback technique described in section 2 can be applied.

5 Experimental Results

To address the challenging issues involved in MIR, a Multimedia Analysis and Retrieval System (MARS) project was started at University of Illinois[6, 11, 12, 13, 14, 15]. MARS-1 is accessible via internet at http:// jadzia.ifp.uiuc.edu:8000. The two relevance feedback procedures discussed in section 4 have been implemented in two subsystems in MARS-2.

5.1 Top layer feedback

This subsystem[14] takes the shape-based image retrieval as an example to illustrate the system's flexibility to support various perception criteria from various users.

Specifically, f_i is the shape feature and (M, R) = {Chamfer, Euclidean, Modified Fourier Descriptor, Hausdorff}. All the elements in (M, R) are invariant to translation, rotation, and scaling. Fast matching algorithms have been developed and implemented[14].

As part of the DLI[18] content-based retrieval test bed, there are about 300 images in the database, which are a collection of ancient African artifacts from the Getty Museum.

An example feedback process is illustrated in Figure 2. (The system is accessible via internet at http://quark. ifp.uiuc.edu:8080) In Figure 2, the upper-left image is the query image. After the query is submitted, the combined rank list l_c is constructed, as described in section 4.1. Retrieved images are then returned to the user in the order specified in l_c. The numbers in the input areas are the combined ranks for the corresponding images. If the user is not satisfied with rank order, he can modify the rank order according to his own judgment. The computer will determine the best (r_j, m_k) according to user's feedback rank list l_f.

According to our intensive tests, we find that all the four elements in (R, M) best match some users' perception. This observation further supports the necessity for MIR to support multi-element (R, M) set.

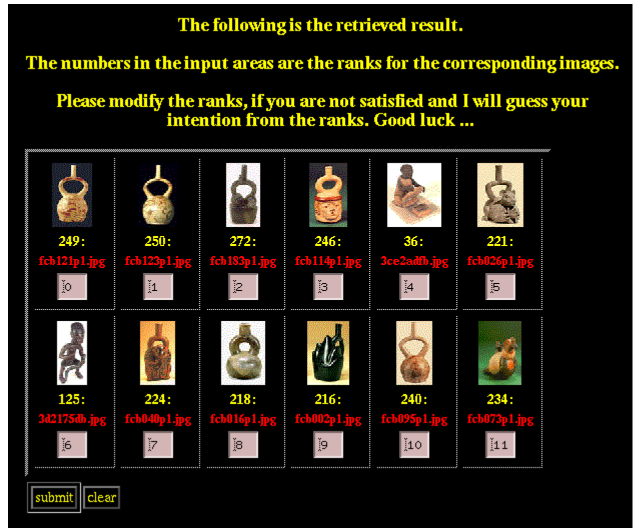

Figure 2: Top Layer Relevance Feedback

Via the top layer relevance feedback, the MIR system is capable of supporting different judgment criteria from different users and thus better meet the user's information need.

5.2 Bottom layer feedback

This subsystem[15] takes the texture-based image retrieval as an example. Specifically, f_i is the texture feature, r_j is wavelet-based representation of texture, and m_k is the Cosine distance.

In wavelet-based texture representation, an image is fed into a wavelet filter bank and is decomposed into de-correlated sub-bands. Each sub-band captures the texture feature of some scale and orientation of the original image. Specifically, we decompose an image into three wavelet levels; thus have 10 sub-bands. For each sub-band, the standard deviation of the wavelet coefficients is extracted. The 10 standard deviations are used as the texture representation for the image.

There are 1888 texture images in the database. The original 118 512×512 images are obtained from MIT Media Lab at ftp://whitechapel.media.mit.edu/ pub /VisTex. Each image is then cut into 16 128×128 non-overlapping small images. The images in the database cover a very wide range of textures, including fabrics, foods, clouds, leaves, barks, buildings, paintings, bricks. Many of them are composed of non-homogeneous textures and thus challenging for retrieval.

An example feedback process is given in Figure 3. (The system is accessible via internet at http://quark. ifp.uiuc.edu:8080) The upper figure is the initial retrieval result with the top-left image being the query image. While some of the retrieval returns are similar to the query image, some, i.e. images 1179, 1256, 86, and 1170, are not good retrieval returns.

By checking the check-boxes under the relevant images, the user feedbacks his judgment to the computer. Based on the user's feedback, the values in v_j of the query is refined[15], as described in sections 2 and 4.2. An improved retrieval result is shown in the lower figure.

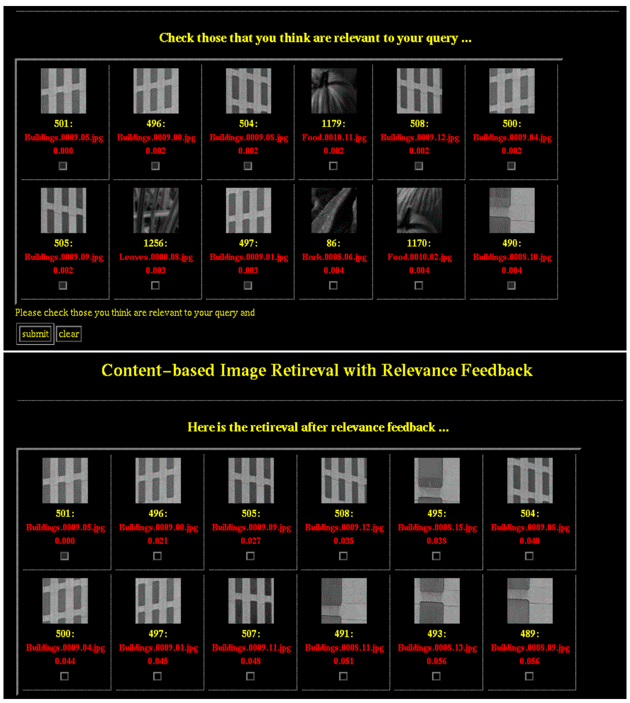

Figure 3: Bottom Layer Relevance Feedback

In the bottom layer relevance feedback, the user is not required to specify the exact values in v_j of his query, which is usually impossible. Instead, started with a coarse initial query, the user only needs to mark the retrieval returns that he thinks is relevant and feedbacks the information to the MIR system. The MIR system is capable of refining the initial query such that the refined query is a more accurate approximation to the user's query in mind; and thus better meet the user's information need.

Currently, these two subsystems are being integrated into a single relevance feedback architecture in MARS-2.

6 Conclusions and Future Work

In the past few years, many retrieval approaches based on extracting and representing visual properties of multimedia data have been developed. While these approaches establish the viability of MIR based on visual features, techniques for incorporating human expertise to improve retrieval performance have not been studied. To address this limitation, this paper introduces a Human-Computer Interaction based approach to MIR in which the user guides the system during retrieval using relevance feedback. Our experiments show that the retrieval performance is considerably improved, in both flexibility and precision, by incorporating humans in the retrieval process.

- With the top-layer relevance feedback, the user is exempt from specifying his perception criterion and the (r_j, m_k) selection is done automatically by the system. This enables the system to have the flexibility to support different perception criteria from different users.
- With the bottom-layer relevance feedback, the user is exempt from specifying an exact query. The initial coarse query will be refined automatically by the system with the user's feedback, such that the refined query is a better approximation to the user's information need.

The integration of relevance feedback into MIR opens a wide research area. We propose the following in our future research.

- *Automatic Feature Mining via Relevance Feedback*: In the paper we assume that a query consists of a single feature and the two-layer relevance feedback is then performed over it. Although this is true in some cases, in some other cases the query can not be captured by a single feature alone. That is, we need multiple features with different weights to specify a query. This problem can be solved in a similar way how automatic (r_j, m_k) selection is done. That is, the alternative procedure of automatic (r_j, m_k) selection discussed at the end of Section 4.1 can be used directly in the automatic feature mining. The interested readers are referred to our research in [19].
- *The Integrated Relevance Feedback in* (F, R, M, V): In this paper, the relevance feedbacks in top-layer $((R, M))$ and in bottom-layer (V) are done *separately*. In addition, as mentioned above that an automatic feature (F) mining process can also be done *separately*. However, we expect an better retrieval performance if the relevance feedback can be applied in all the three layers *simultaneously*. Our preliminary research results show that this is a promising research direction [19].

References

[1] J. P. Callan, W. B. Croft, and S. M. Harding, "The inquery retrieval system," in *Proc of 3rd Int Conf on Database and Expert System Application.*

[2] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, 1983.

[3] J. R. Smith and S.-F. Chang, "Tools and techniques for color image retrieval," in *IS & T/SPIE proceedings Vol.2670, Storage & Retrieval for Image and Video Databases IV.*

[4] A. Pentland, R.W.Picard, and S.Sclaroff, "Photobook: Tools for content-based manipulation of image databases," in *Proc. Storage and Retrieval for Image and Video Databases II*, vol. 2, (Bellingham, Wash), pp. 34–47, 1994.

[5] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," tech. rep., CIPR TR-95-06, 1995.

[6] T. S. Huang, S. Mehrotra, and K. Ramchandran, "Multimedia analysis and retrieval system (MARS) project," in *Proc of 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval*, 1996.

[7] M. Flickner, H. Sawhney, W. Niblack, and J. Ashley, "Query by image and video content: The qbic system," *IEEE Computer*, 1995.

[8] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. fe Shu, "The virage image search engine: An open framework for image management," in *SPIE Storage and Retrieval for Still Image and Video Databases IV.*

[9] W. M. Shaw, "Term-relevance computations and perfect retrieval performance," *Information Processing and Management.*

[10] T.P.Minka and R.W.Picard, "Interactive learning using a "society of models"," in *Proc IEEE 1996 CVPR'96.*

[11] S. Mehrotra, Y. Rui, M. Ortega-B., and T. S. Huang, "Supporting content-based queries over images in MARS," in *Proc. of IEEE Int. Conf. on Multimedia Computing and Systems*, 1997.

[12] Y. Rui, A. C. She, and T. S. Huang, "Modified fourier descriptors for shape representation – a practical approach," in *Proc of First International Workshop on Image Databases and Multi Media Search.*

[13] Y. Rui, A. C. She, and T. S. Huang, "Automated shape segmentation using attraction-based grouping in spatial-color-texture space," in *Proc. IEEE Int. Conf. on Image Proc.*, 1996.

[14] Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "Automatic matching tool selection using relevance feedback in MARS," in *(submitted to) Inter. Conf. on Visual Information Retrieval*, 1997.

[15] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in *Proc. IEEE Int. Conf. on Image Proc.*, 1997.

[16] Y. Rui, K. Chakrabarti, S. Mehrotra, Y. Zhao, and T. S. Huang, "Dynamic clustering for optimal retrieval in high dimensional multimedia databases," in *(submitted to) Int. Conf. on Data Engineering*, 1997.

[17] C. Buckley and G. Salton, "Optimization of relevance feedback weights," in *Proc. of SIGIR'95.*

[18] B. Schatz and H. Chen, "Building large-scale digital libraries," *Computer*, 1996.

[19] Y. Rui, T. S. Huang, and S. Mehrotra, "An integrated relevance feedback architecutre in multimedia information retrieval," in *IFP Technical Report YR06-97, Beckman Institute, UIUC*, 1997.