A METHOD TO ASSOCIATE USER SUBJECTIVITY TO IMAGE DATABASES IN CBIR

Author

Dr. PUNKAJ JALVI
Research Scholar
Magadh University, Bodh Gaya.

ABSTRACT

Image retrieval is an important topic in the field of pattern recognition and artificial intelligence. There are three categories of image retrieval methods: text-based, content-based and semantic-based. In Content-Based Image Retrieval (CBIR), images are indexed by their visual content, such as color, texture, shapes. CBIR has become one of the most active research areas in the past few years. Many visual feature representations have been explored and many systems are built. While these research efforts are established the basis of CBIR, the usefulness of the proposed approaches is limited. Specially, these efforts have relatively ignored two distinct problems of CBIR systems: The semantic gap between high level concepts and low level features; Human perception of visual content. In addition to this, we have the problem of which image analysis models to use in image database to achieve a better CBIR system.

This paper proposes a novel method for combining the user subjectivity in image database and interactive content-based image retrieval (CBIR). It shows a two-step process: Performs image analysis before retrieving an image from the database, which automatically infers which combination of models best are to represents the data of interest to the user and learns continuously during interaction with each user. Effectively takes the above two problems into account in CBIR. In the retrieval process, the user's high level query and perception subjectivity are captured by dynamically updated weights based on the user's feedback. The proposed approach greatly reduces the user's effort of composing a query and captures the user's information.

Keywords: Content-Based Image Retrieval, Relevance Feedback, Image Database and Image Analysis.

1. INTRODUCTION

With advances in the computer technologies and the advent of the World-Wide Web, there has been an explosion in the amount and complexity of digital data being generated, stored, transmitted, analyzed, and accessed. Much of this information is multimedia in nature, including digital images, video, audio, graphics, and text data. In order to make use of this vast amount of data, efficient and effective techniques to retrieve multimedia information based on its content need to be developed. Not only it is the most widely used media type besides text, but it is also one of the most widely used bases for representing and retrieving videos and other multimedia information.

The retrieval of images based on their contents, even though the approach is readily generalizable to other media types. Keyword annotation is the traditional image retrieval paradigm. In this approach, the images are first annotated manually by keywords. They can then be retrieved by their corresponding annotations. However, there are three main difficulties with this approach, i.e. the large amount of manual effort required in developing the annotations, the differences in interpretation of image contents, and inconsistency of the keyword assignments among different indexers [1], [2], [3]. To overcome the difficulties of the annotation based approach, an alternative mechanism, Content-Based Image Retrieval (CBIR), has been proposed in the early 1990's. Besides using human-assigned keywords, CBIR systems use the visual content of the images, such as color, texture, and shape features, as the image index. This greatly alleviates the difficulties of the pure annotation based approach, since the feature extraction process can be made automatic and the image's own content is always consistent. CBIR has attracted great research attention, ranging from government [4], [5], industry [2], [6], [7], to universities [8], [9], [10], [11], [12]. The earliest systems designed for image retrieval and those that have become commercially available, tend to follow a basic paradigm: (1) pre compute features or model parameters for each image (2) have the user specify which models or ranges of parameters are most important, and (3) have the user select example images to initiate a query. The system then compares the user's query information with all the stored information, and retrieves images it thinks are “similar" according to the constraints specified during step (2).

This basic paradigm is useful in limited data sets and search problems, provided that the user is an expert in how the underlying image similarity processing works. However, it is not suitable for general use. The average person looking for images does not know how to choose model parameters as required in step (2). Moreover, as combinations of models (e.g. multiple color and texture models) become available, the choice of parameters is non-intuitive even for the expert image processing researcher. In short, new image analysis tools are needed that perform model selection and combination.

The corresponding system design strategy for early CBIR systems is to first and the best representations for the visual features. Then:

- During the retrieval process, the user selects the visual feature(s) that he or she is interested in. In the case of multiple features, the user needs to also specify the weights for each of the features.

- Based on the selected features and specified weights, the retrieval system tries to and similar images to the user's query.

- We refer to such systems as computer centric systems. While this approach establishes the basis of CBIR, the performance is not satisfactory due to the following two reasons:

- The semantic gap between high level concepts and low level features.

The assumption that the computer centric approach makes is that the high level concepts to low level features mapping is easy for the user to do. While in some cases the assumption is true, e.g. mapping a high level concept (fresh apple) to low level features (color and shape), in other cases, this may not be true. One example is to map an ancient vase with sophisticated design to an equivalent representation using low level features. The gap exists between the two levels.

- The subjectivity of human perception.

Different persons, or the same person under different circumstances, may perceive the same visual content differently. This is called human perception subjectivity [13]. The subjectivity exists at various levels. For example, one person may be more interested in an image's color feature while another may be more interested in the texture feature. Even if both people are interested in texture, the way how they perceive the similarity of texture may be quite different.

Additionally, a user should be allowed to be subjective to give, over time, the same set of imagery different labels, or to give the same labels to different content, e.g. to the category of images “they like." It is desirable that the system be able to adapt itself continuously to the changing requests of the user, e.g. to *learn* how to model mappings between the image data and its labels based on changing feedback from the user.

One of the most challenging test scenarios is when the desired image contents are hard to describe objectively. In the solutions we are researching, the users do not have to select model parameters, but simply choose example images that they like. Fig. 1 illustrates a case of two users trying to find more images they like in a Picasso art database of 320 paintings. Each user selects a few example images, and then the system analyzes the characteristics of the examples and retrieves other similar images from the database. The burden is on the system to infer how to measure similarity.

In the Fig.1. User 1 gives two examples of textured, cubist paintings of different colors. The system infers that color is not relevant, and searches for images with similar texture (using a multiscale simultaneous autoregressive texture model from [14]). User 2 also gives two examples. The first image is identical to that of user 1, but the second has a different texture and the same color. In this case, the system determines that color is important and retrieves other images with similar colors (using Euclidean distances on 256-bucket color histograms from the decorrelating color space of [15]). The browser can also combine texture and color for one query, or choose combinations of other available similarity models. To refine the query results, the user simply gives additional examples. This is called “relevance feedback". Image analysis aims to infer underlying features common to the user-defined category, and then use these features to predict other images of interest to the user. The inference is always based on the present set of positive and negative examples given by a user. It is important to note that modeling subjectivity is an objective problem. The labeling or categories chosen by subjective users result in objective groupings of data over which the performance of the system can be tested objectively.

The rest of the paper is organized as follows. Section 2 contains related work, Section 3 contains the proposed system, describes a method of inference for models and their combination, describes user subjectivity procedure, a multimedia object model which

supports multiple features, multiple representations, and their corresponding weights, The weights are essential in modeling high level concepts and perception subjectivity and discusses how the weights are dynamically updated based on the relevance feedback to track the user's information need. Section 4 contains conclusion of the paper.

2. RELATED WORK

In the computer centric approach, the “best" features and representations and their corresponding weights are fixed, which cannot effectively model high level concepts and user's perception subjectivity. Furthermore, specification of weights imposes a big burden on the user, as it requires the user to have a comprehensive knowledge of the low level feature representations used in the retrieval system, which is normally not the case. Motivated by the limitations of the computer centric approach, recent research focus in CBIR has moved to an interactive mechanism that involves a human as part of the retrieval process [16], [17], [18], [19]. Exampleinclude interactive region segmentation [20]; interactive image database annotation [18], [21]; usage of supervised learning before the retrieval [22], [23]; and interactive integration of keywords and high level concepts to enhance image retrieval performance [24].

In this paper, to address the difficulties faced by the computer centric approach, we present a Relevance Feedback based approach to CBIR, in which human and computer interact to refine high level queries to representations based on low level features. Relevance feedback is a powerful technique used in traditional text-based Information Retrieval systems. It is the process of automatically adjusting an existing query using the information fed-back by the user about the relevance of previously retrieved objects such that the adjusted query is a better approximation to the user's information need [25], [26], [27]. In the relevance feedback based approach [17], [28], [29], [13], the retrieval process is interactive between the computers and human. Under the assumption that high-level concepts can be captured by low-level features, the relevance feedback technique tries to establish the link between high-level concepts and low-level features from the user's feedback. Furthermore, the burden of specifying the weights is removed from the user. The user only needs to mark which images he or she thinks are relevant to the query. The weights embedded in the query object are dynamically updated to model the high level concepts and perception subjectivity.

The retrieval techniques used in CBIR systems lag behind the corresponding techniques in today's best text search engines, such as Inquery [30], Alta Vista, Lycos, etc. At the early stage of CBIR, research primarily focused on exploring various feature representations, hoping to and a “best" representation for each feature. For example, for the texture feature alone, almost a dozen representations have been proposed [16], including Tamura [31], MSAR [32], Word decomposition [33], Fractal [34], Gabor Filter [35], [11], and Wavelets [36], [37], [12], etc.

3. PROPOSED SYSTEM

Fig.2. shows the general scheme of the proposed system using relevance feedback in CBIR. The basic idea of relevance feedback is to shift the burden of finding the right query formulation from the user to the system.

Algorithm for proposed system is as follows:

Step1: The initially the image database will be supplies to the system, which is a raw database not applied any technique on it (i.e. contains only set of images).

Step 2: This database is supply to model inference and combination; it identifies the best model for solving the problem.
Step 3: Learning and Generalization process will do to find the user subjectivity.

Step 4: Then modifies the initial database. Now it is ready for retrieval process, user will gives the query image to this modified image database.

Step 5: Before entering into the CBIR system, it creates object model—formalizes how an image object is modeled.
Step 6: Entered into the relevance feedback system

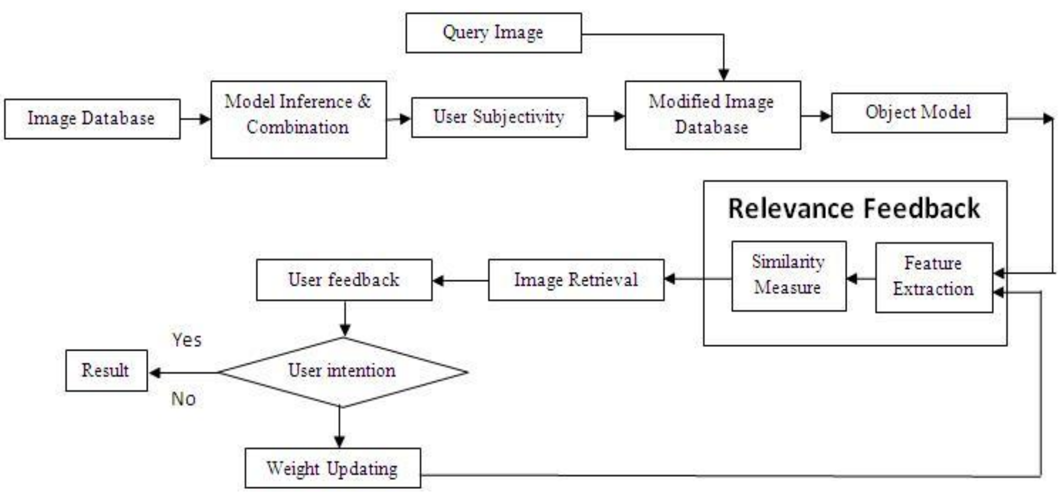


Fig.2. Proposed System Architecture

Step 7: after the step 6, system is retrieves the set of images for given query image on which user is going to give feedback in the way to find the relevant or irrelevant images.

Step 8: If the feedback is YES, then displays set of images of user required. If NO, adjust the weights and return back to the Step 6.
Step 9: Repeat the steps from 6 to 8, until user gets relevant images.

The following sections are describes in detail about important parts of above proposed system. They are as follows:

3.1. MODEL INFERENCE AND COMBINATION

For years many researchers have assumed that there would be “one best" model for solving the problem of perceptual similarity, or image similarity. Working in the area of content-based image retrieval has changed our thinking in this regard. Although there are searches on limited domains where one model may always be best, in general we think the one-model solution will be too brittle, and a relatively small set of models (less than a dozen) will give the best performance.

Of course, which models these should be remains dependent on the data and what is to be done with it. In the Picasso example above, the particular texture model is good at grouping some of Picasso's cubist paintings, but poor at grouping portraits. Another model, or combination of models, might perform better still. With no claims of starting with the best models, but with evidence that combining suboptimal ones can outperform a single one [26], we describe the following method for making combinations which can improve joint performance. We have explored many ways for combining models.

Initially, we considered direct linear combinations of model features--the traditional approach. However, concatenating model features causes an exponential growth in the space used to represent the data, and has a variety of other problems, especially the problem when features from one model are of a different scale than features from another model, and simple re-scaling of them destroys their discrimination properties [27]. To date, the most successful combination method we have found (for avoiding the scaling and dimensionality problems, and for running in interactive time) is based on quantization of the feature spaces followed by a learning algorithm, such as set cover [26].

The model features or parameters are used only initially during quantization, which is represented as hierarchical trees. With the tree representation, different segmentations can be made by simply choosing a different partition of the nodes. The hierarchical trees are representations that make segmentations, as opposed to a representation that is a fixed segmentation. Once the user is in the system loop, the system can decide which of its possible segmentations best suits the user's desires. Once the trees are constructed, the similarity problem changes from one of metric distances on image features to one of distances in a hierarchy-induced metric. Different model parameter ranges and dimensionalities cease to be an issue. The set-cover combination method then proceeds by looking for the simplest set of nodes that covers all the user's positive examples and none of their negative examples. An example of using this method, starting with three models (and a database of only six elements) is shown in Fig.3. In Fig.3, each model constructs a tree of possible segmentations. The user provides examples: (a, b, c, d are positive, e is negative.) The system chooses tree nodes which most efficiently describe the positives and not the negatives, resulting in the two groupings shown (shaded). The result is a combination of color and texture to characterize the user's examples. Note that example f is inferred as being of interest to the user.

The inference method presently acts on both positive and negative examples. A limitation is that the user cannot yet give feedback such as, “I don't like this particular spatial arrangement of these colors." This is an area for continued development.

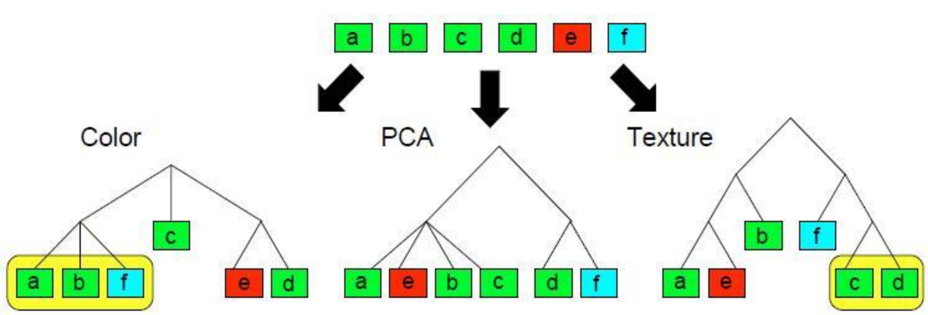


Fig.3. Method for combining multiple models.

The current inference or combination method processes about 5 examples per CPU second on an HP 735/99 using a database with thousands of leaf elements and about a half dozen models. Details on its complexity are in [38]. While we generally don't like to impose speed requirements on research algorithms, the ability to perform model selection interactively is a benefit in a retrieval system, where the formulation of a query is well-suited to an iterative process of providing a few examples, seeing what is retrieved, modifying the example set, and so forth. This ability obviates the need for the user to “think of everything" before posing a query. In reality, users often modify what they want after seeing more of the database.

3.2. LEARNING AND GENERALIZATION- MODELING USER SUBJECTIVITY

Most current retrieval systems, however, have no such memory. When people repeatedly ask similar queries of them, the system appears to be “stupid" because it doesn't learn. Therefore, on top of the set-covering algorithm, we've added a dynamic bias. The *bias* of a learner is defined to be any basis for choosing one generalization over another, other than strict consistency with the training examples [42]. Having the right bias is crucial to successful learning, especially when a small number of examples (as desired in an interactive setting) leave open many possible solutions.

The “FourEyes" browser improves its bias over time [38]. When the system sees a problem similar to one it has seen before, it automatically switches to the bias that it learned for that problem. When it sees a significantly new problem, FourEyes learns a new bias. It therefore behaves differently over time, depending on what it has been exposed. FourEyes has three stages that learn at different rates, from interactive-speed online learning, to longer-term offline learning, the latter of which is analogous to human “reflection" or “dreaming" in its abilities to process image information over a broader scope.

One of the critical tests of a learning system is how well does it generalize? Traditional image processing has been concerned with generalization from a training set to a test set. The problem in image retrieval systems is that the same test set might have more than one “true" interpretation, depending on what the user wants at the moment.

The user's subjectivity, in the form of changing feedback, creates a signal processing problem analogous to non-stationary signal detection, where the category of signals you are trying to detect, e.g., “images you like," may change its signature in time. But they also may not change. The difficulty is to track the changes, while preserving performance on the parts that do not change. The system tracks the changes with online clustering of the bias, represented as weights on the tree nodes. (These weights are used in the set-cover process.) Another aspect of the bias is the shape of the trees, corresponding to the quantization of the space. This is also learned during interaction with the user.

3.3. THE MULTIMEDIA OBJECT MODEL

Before we describe how the relevance feedback technique can be used for CBIR, we first need to formalize how an image object is modeled [13]. An image object O is represented as:

$$= O\ (D, F, R) \tag{1}$$

- D is the raw image data, e.g. a JPEG image.
- $F = \{f_i\}$ is a set of low-level visual features associated with the image object, such as color, texture, and shape.
- $R = f\{r_{ij}\}$ is a set of representations for a given feature fi , e.g. both color histogram

and color moments are representations for the color feature [40]. Note that, each representation r_{ij} itself may be a vector consisting of multiple components,

$$r_{ij} = [r_{ij1}, \ldots r_{ijk}, \ldots\ldots r_{ijK}] \tag{2}$$

where K is the length of the vector.

In contrast to the computer centric approach's single representation and fixed weights, the proposed object model supports multiple representations with dynamically updated weights to accommodate the rich content in the image objects. Weights exist at various levels. W_i, W_{ij} , and W_{ijk}, are associated with features f_i, representations r_{ij} , and components rijk, respectively. The goal of relevance feedback, described in the next section, is to and the appropriate weights to model the user's information need. Further, note that a query Q has the same model as that of the image objects, since it is also an image object in nature.

3.4. INTEGRATING RELEVANCE FEEDBACK IN CIBR

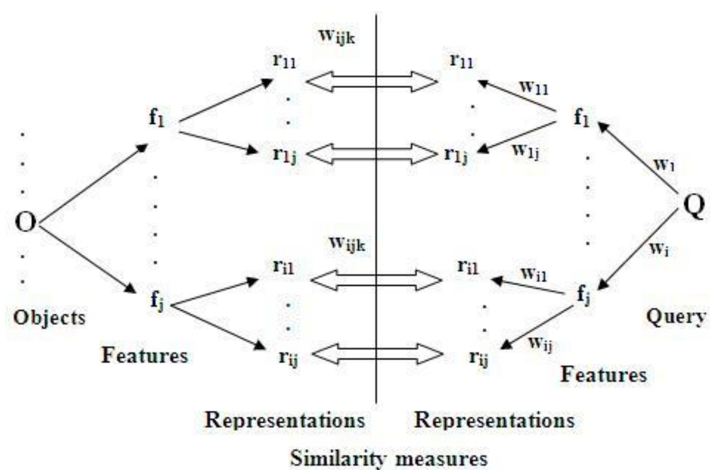
An image object model O(D,F,R) together with a set of similarity measures M = $\{m_{ij}\}$, specifies a CBIR model (D,F,R,M). The similarity measures are used to determine how similar or dissimilar two objects are. Different similarity measures may be used for different feature representations. For example, Euclidean is used for comparing vector-based representations, while Histogram Intersection is used for comparing color histogram representations. Based on the image object model and the set of similarity measures, the retrieval process is described below and also illustrated in Fig.4.

1. Initialize the weights W = $[W_i, W_{ij}, W_{ijk}]$ to W_0, which is a set of no-bias weights. That is, every entity is initially of the same importance

$$'_i = W0_i = 1/I \qquad (3)$$

$$'_{ij} = W0_{ij} = 1/J_i \qquad (4)$$

$$'_{ijk} = W0_{ijk} = 1/K_{ij} \qquad (5)$$



where I is the number of features in set F; Ji is the number of representations for feature f_i; K_{ij} is the length of the presentation vector r_{ij} .

2. The user's information need, represented by the query object Q, is distributed among different features f_i, according to their corresponding weights W_i.

3. Within each feature f_i, the information need is further distributed among different feature representations r_{ij} , according to the weights W_{ij} .

4. The objects' similarity to the query, in terms of r_{ij}, is calculated according to the corresponding similarity measure m_{ij} and the weights W_{ijk}:

$$r_{ij}) = m_{ij}(r_{ij}, W_{ijk}) \qquad (6)$$

5. Each representation's similarity values are then combined into a feature's similarity value:

$$f_i) = \sum W_{ij} S(r_{ij}) \quad (7)$$

The overall similarity S is obtained by combining individual

$$= \sum W_i S(f_i) \quad (8)$$

7. The objects in the database are ordered by their overall similarity to Q. The NRT most similar ones are returned to the user, where NRT is the number of objects the user wants to retrieve.
8. For each of the retrieved objects, the user marks it as highly relevant, relevant, no-opinion, non-relevant, or highly non-relevant, according to his information need and perception subjectivity.
9. The system updates the weights according to the user's feedback such that the adjusted Q is a better approximation to the user's information need.
10. Go to Step 2 with the adjusted Q and start a new iteration of retrieval.

In Fig.4, the information need embedded in Q flows up while the content of O's flows down. They meet at the dashed line, where the similarity measures m_{ij} are applied to calculate the similarity values $S(r_{ij})$'s between Q and O's.

Following the Information Retrieval theories [26], [26], [27], the objects stored in the database are considered objective and their weights are fixed. Whether the query is considered objective or subjective and whether its weights can be updated distinguishes the proposed relevance feedback approach from the computer centric approach. In the computer centric approach, a query is considered objective, the same as the objects stored in the database, and its weights are fixed. Because of the fixed weights, this approach cannot effectively model high level concepts and human perception subjectivity. It requires the user to specify a precise set of weights at the query stage, which is normally not possible. On the other hand, queries in the proposed approach are considered as subjective. That is, during the retrieval process, the weights associated with the query can be dynamically updated via relevance feedback to react the user's information need. The burden of specifying the weights is removed from the user.

Note that in the proposed retrieval algorithm, both S and $S(f_i)$ are linear combinations of their corresponding lower level similarities. The basis of the linear combination is that the weights are proportional to the entities' relative importance [41]. For example, if a user cares twice as much about one feature (color) as he does about another feature (shape), the overall similarity would be a linear combination of the two individual similarities with the weights being 2/3 and 1/3, respectively[41]. Furthermore, because of the nature of linearity, these two

vels can be combined into one, i.e.

$$= \sum_{j} \sum W_{ij} S(r_{ij}) \quad (9)$$

where W_{ij} 's are now re-defined to be the weights by which the information need in Q is distributed directly into r_{ij} 's. Note that it is not possible to absorb W_{ijk} into W_{ij} , since the calculation of $S(r_{ij})$ can be a non-linear function of W_{ijk}'s, such as Euclidean or Histogram Intersection.

In the next two sections, we will discuss two key components of this retrieval algorithm, i.e. normalization and weight updating.

3.5. WEIGHT UPDATING

After the intra and inter normalization procedures discussed above, the components r_{ijk} within a vector r_{ij}, as well as $S(r_{ij})$'s within the overall similarity S, are of equal emphasis. This objective equality allows us to meaningfully associate subjectively unequal intra and inter weights for a particular query.

3.5.1. UPDATE OF W_{ij} (INTER-WEIGHT)

The W_{ij}'s associated with the r_{ij}'s react the user's different emphasis of a representation in the overall similarity. The support of different weights enables the user to specify his or her information need more precisely. We will
next discuss how to update W_{ij}'s according to user's relevance feedback.

3.5.2. UPDATE OF W_{ijk} (INTRA-WEIGHT)

The W_{ijk}'s associated with r_{ijk}'s react the different contributions of the components to the representation vector r_{ij}. For example, in the wavelet texture representation, we know that the mean of a sub-band may be corrupted by the lighting condition, while the standard deviation of a sub-band is independent of the lighting condition. Therefore more weight should be given to the standard deviation component, and less weight to the mean component. The support of different weights for r_{ijk}'s enables the system to have more reliable feature representation and thus better retrieval performance.

3.6. SUMMARY

Based on the description of the relevance feedback algorithm in Sections 3.4, 3.5 and 3.6 we briefly summarize the properties of the algorithm.

- Multi-modality:

The proposed image object model, and therefore the retrieval model, supports

multiple features and multiple representations. In contrast to a computer centric approach's attempt of finding the single “best" universal feature representation, the proposed approach concentrates on how to organize the multiple feature representations, such that appropriate feature representations are invoked (emphasized) at the right place and right time. The multi-modality approach allows the system to better model user's perception subjectivity.

- Interactivity

In contrast to a computer-centric approach's automated system, the proposed approach

is interactive in nature. The interactivity allows the system to make use of the ability both from computer and from human.

- Dynamic

In contrast to a computer-centric approach's fixed query weights, the proposed

approach dynamically updates the query weights via relevance feedback. The advantages are twofold:

Remove burden from the user

The user is no longer required to specify a precise set of weights at the query formulation stage. Instead, the user interacts with the system, indicating which returns he or she thinks are relevant. Based on the user's feedback, query weights are dynamically updated.

Remove burden from the computer

The computer is no longer required to understand the high level concept. Based on user's feedback, the high level concept embedded in the query weights automatically gets refined.

4. CONCLUSION

This paper has highlighted results from our recent research focusing on image analysis for (1) model inference and combination, and (2) learning for generalization. The underlying premise is that a subjective human is in the loop with the image analysis and retrieval system. Not only does the human user unlikely to know how to set all the model parameters optimally, but his or her subjectivity leads to the same data need to be treated in different ways. A method was described for automatically choosing combinations from multiple image models. This releases the human from the task of adjusting image features or model parameters. This is intended to simulate training after interacting with one user, and then having to perform well while working with another user or with the same user who is behaving differently over time. To a user, this behavior would make the system appear “smarter and faster" with increasing use.

CBIR has emerged as one of the most active research areas in the past few years. Most of the early research effort focused on finding the “best" image feature representations. Retrieval was performed as summation of similarities of individual feature representation with fixed weights. While this computer centric approach establishes the basis of CBIR, the usefulness of such systems was limited due to the difficulty in representing high level concepts using low level features and human perception subjectivity. In this paper, we introduce a Human-Computer Interaction approach to CBIR based on relevance feedback. Unlike the computer centric approach, where the user has to precisely decompose his information need into different feature representations and precisely specify all the weights associated with them, the proposed interactive approach allows the user to submit a coarse initial query and continuously refine his information need via relevance feedback.

This approach greatly reduces the user's effort of composing a query and captures the user's information need more precisely. Furthermore, the efficiency and effectiveness of the proposed approach have been validated by a large amount of experiments.

REFERENCES

[1] W. Niblack, R. Barber, and et.al.: The QBIC project: Querying images by content using color, texture and shape. in Proc. SPIE Storage and Retrieval for Image and Video Databases, Feb 1994.

[2] M. Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Q. Huang, Byron Dom, Monika Gorkani, Jim Ha_ne, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. : Query by image and video content: The QBIC system. IEEE Computer, 1995.

[3] C. Faloutsos, M. Flickner, W. Niblack, D. Petkovic, W. Equitz, and R. Barber.: Efficient and effective querying by image content. Tech. Rep. IBM Research Report, 1993.

[4] R. Jain.: Workshop report: NSF workshop on visual information management systems. in Proc. SPIE Storage and Retrieval for Image and Video Databases,1993.

[5] R. Jain, A. Pentland, and D. Petkovic.: NSF-ARPA workshop on visual information management systems.Cambridge, MA, June 1995.

[6] Jeffrey R. Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh Jain, and Chiao fe Shu.: The Virage image search engine: An open framework for image management. in Proc. SPIE Storage and Retrieval for Image and Video Databases.

[7] James Dowe.: Content-based retrieval in multimedia imaging. in Proc. SPIE Storage and Retrieval for Image and Video Databases, 1993.

[8] A. Pentland, R. W. Picard, and S. Sclaro.: Photobook: Content-based manipulation of image databases. International Journal of Computer Vision, 1996.

[9] Thomas S. Huang, Sharad Mehrotra, and Kannan 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.

[10] J. R. Smith and S.-F. Chang.: Visually searching the web for content. IEEE Multimedia Magazine, vol. 4, no. 3, pp. 12-20,Summer 1997, also Columbia U. CU/CTR Technical Report 459-96-25.

[11] W. Y. Ma and B. S. Manjunath.: Netra: A toolbox for navigating large image databases. in Proc. IEEE Int. Conf. on Image Proc., 1997.

Conflict of Interest Reported: Nil; Source of Funding: None Reported.