Proceedings of the 2003 Systems and Information Engineering Design Symposium
Matthew H. Jones, Barbara E. Tawney, and K. Preston White, Jr., eds.

A DECENTRALIZED IMAGE RETRIEVAL SYSTEM FOR EDUCATION

Jinshan Tang

Virginia Image and Video Analysis
Department of Electrical and Computer Engineering
University of Virginia, VA 22903, U.S.A.

Scott T. Acton

Virginia Image and Video Analysis
Department of Electrical and Computer Engineering
University of Virginia, VA 22903, U.S.A.

ABSTRACT

In this paper, we describe the development of a decentralized image retrieval system for education (DIRECT). With DIRECT, content based image retrieval (CBIR) is a service that will run autonomously; the constituent collections will not need to be modified to accommodate DIRECT. DIRECT applies a consistent CBIR mechanism to the entire system and gives CBIR functionality to collections without effort by the collection providers. Collections that do not have their own CBIR system or are not able to invest in a CBIR system (such as small collections or collections that do not contain a large proportion of images) will gain greater distribution of their holdings via DIRECT's CBIR functionality. The DIRECT service finds the collections in the library and requests the images using the same method that the portal uses to search and request images. DIRECT will perform feature extraction on the images and store the features in its own database. A portal or client application can query DIRECT, and it will use the features of the desired image to search the library for images with the same features. DIRECT will automatically monitor collections in order to keep the DIRECT index up-to-date.

1 INTRODUCTION

In order to support education in science, technology, engineering, and mathematics, the National Science Foundation has established the National Science, Technology, Engineering, and Mathematics Education Digital Library (NSDL) program [NSF-02-054. 2002].

One challenging research problem that needs to be addressed in the NSDL research community is mining images in a digital library. The most common method of searching for images involves matching a text query to text metadata that have been entered manually. Text based searches often prove ineffectual for the following reasons: (1) The image does not have associated metadata; (2) metadata are incomplete; (3) metadata include technical terms unknown to the user; (4) subjectivity of metadata causes mismatch between cataloger and user. To assist users in finding desired images from the expected tens of millions of images in the NSDL, innovative services must be pursued to increase the usefulness of image retrieval for both teachers and students. Content based image retrieval (CBIR) techniques can be designed to meet this aim.

In the past years, Content based image retrieval (CBIR) has found wide application in different areas, such as medicine, architectural and engineering design, weather forecasting and so on. A number of CBIR systems have been proposed. Examples of these systems include MIT's Photobook system [Pentland *et al.* 1996], Columbia's VisualSEEK system [Smith *et al.* 1996], UCSD's Virage™ [Bach *et al.* 1996] and IBM's QBIC™ system [Flickner *et al.* 1995]. However, in each case, the search and retrieval of images is based on a central database. These systems place the responsibility of extracting and indexing features on the collections and on the digital librarians. Centralized image retrieval systems cannot meet the needs of NSDL as the NSDL is a heterogeneous collection of distributed resources.

In this paper, we developed a new system which is suitable for the NSDL. Our system, DIRECT, is a decentralized image retrieval system. The DIRECT service does not modify existing collections nor does it require existing collections to meet new standards. DIRECT applies a consistent CBIR mechanism to the entire system and gives CBIR functionality to collections without effort by the collection providers. Collections that do not have their own CBIR system or are not able to invest in a CBIR system (such as small collections or collections that do not contain a large proportion of images) will gain greater distribution of their holdings via DIRECT's CBIR functionality.

The rest of the paper is organized as follows: in the second section, we will describe the mechanism of content based image retrieval in DIRECT. Feature extraction, similarity measurement, feature fusion and relevance feedback will be discussed in this section. In Section 3, we will focus the description on the peer to peer file retrieval in DIRECT. Finally, a conclusion will be given in Section 4.

2 CONTENT BASED IMAGE RETRIEVAL IN DIRECT

2.1 Feature Extraction

Color and texture are the main features currently used in DIRECT.

For color description, we exploit color histograms. Because of the robustness to image size and orientation, the color histogram is the one of the most commonly used representations in CBIR. For each image, DIRECT records a quantized (16 element) histogram for each color band. Although perceptually motivated color spaces exist, such as l*a*b*, we utilize the RGB color space to achieve enhanced processing versatility (see Fig. 1 for example). If we required an l*a*b* representation for color, the majority of images in the distributed database could not meet the specification, due to lack of information about the "true" color content. The color histograms used in our CBIR system are quantized using a preset table that exploits the color discrimination of the human visual system [Ma and Manjunath 1997]. Several images in the database will not have chrominance information but will only have luminance information, as is the case with grayscale images. In this case, matching will occur based only on the luminance (intensity) information.

Figure 1. An image and its color histograms. The histograms are shown using the RGB color space.

For texture extraction, we want to describe the granularity (periodicity) and the directionality of two-dimensional patterns in the digital library images, so that users can retrieve images containing similar patterns (textures). The established methods of graylevel co-occurrence matrices [Haralick and Bosley 1973] and Gabor filter decomposition [Bovik *et al.* 1990] are insufficient for the CBIR task at hand. Co-occurrence matrices are computationally expensive, too sensitive to local changes in texture and do not scale to coarser patterns. The Gabor filter approach allows textures of a variety of scales, but suffers from poor boundary localization and complex filter bank design problems. Our texture description method is based on the wavelet transform [Do and Vetterli 2002].

Wavelet based texture retrieval is founded on the assumption that the energy distribution in the frequency domain can be used to capture texture. Let an image be decomposed into K subbands. Then the energy in the i-th subband can be computed as

$$E_i = \sqrt{\frac{\sum_{j=1}^{K_i} x_{i,j}^2}{K_i}} \qquad (1)$$

where $x_{i,1}, x_{i,2}, \ldots, x_{i,K_i}$ are the wavelet coefficients in the i-th subband and K_i is the number of the wavelet coefficients in the i-th subband. Using the energy in K subbands, we can obtain a feature vector representing the texture characteristics of the image

$$E = (E_0, E_1, \ldots, E_{K-1}) \qquad (2)$$

In this paper, the multiscale decomposition technique developed in [Mallat 1989] is used to decompose the images into different subbands. In this technique, first the original image is decomposed into four subbands: the low–low subband, the low-high subband, the high-low subband and the high-high subband; then, the low–low subband is further decomposed into four other subbands similar to the subbands obtained by the original image. This decomposition can be carried out further as necessary. In DIRECT, we employ three levels of decomposition.

2.2 Similarity Measure

After we extract the features of the images, we need a measure to compare the similarity between the query image and the image in the database.

For measuring the color similarity between two images, we use the histogram intersection method developed in [Swain and Ballard 1991]. Let I_Q denote the query image, I_D be an image in the database. The similarity between the two images measured by histogram feature is obtained by

$$D_H(I_Q, I_D) = \frac{\sum_{j=0}^{n-1} \min(H(I_Q, j), H(I_D, j))}{\sum_{j=0}^{n-1} H(I_D, j)} \qquad (3)$$

where $\{H(I_Q, j)\}$ and $\{H(I_D, j)\}$ are the histograms of query image and image from the database respectively.

For measuring the texture similarity between two images, we use Euclidean distance:

$$D_T(I_Q, I_D) = \sqrt{\sum_{j=0}^{K-1} (E_j - E_j^{'})^2} \qquad (4)$$

where $\{E_j\}, \{E_j^{'}\}$ (j=0,1,...,K-1) represent the feature vector obtained using query image and the image in the database respectively.

2.3 Feature Fusion

Sometimes, combining two features can lead to improved results over only using a single feature. To combine two features in the image retrieval, a matching functional combining two features is required. The matching functional determines the "goodness" of match between the query image specified by the student/teacher and an image in the distributed digital library. The first step in constructing the matching functional is normalization of the color and texture features for each image. In each segment, the quantized RGB color histograms are normalized, as are the vectors representing texture. Given two images, I_Q and I_D, the matching functional is computed using

$$D(I_Q, I_D) = w_1 \overline{D}_H(I_Q, I_D) + w_2 \overline{D}_T(I_Q, I_D) \qquad (5)$$

where $\overline{D}_H(I_Q, I_D)$ are the normalized quantized RGB color histograms similarity and $\overline{D}_T(I_Q, I_D)$ represent the normalized similarity of texture. The parameters w_1, w_2 determine the relative weighting for the image features. In the initial deployment of the DIRECT system, we allow only discrete weights of {0, 1, 2, 3}. A weight of zero would exclude a particular feature (e.g., texture) from the query. A weighting using 1, 2 and 3 would assign a prioritization of features in the search. When all weights are equal (i.e., =1), the characteristics of color and texture are considered equally in the image search.

2.4 Relevance Feedback

Another way to improve the retrieval performance is to use relevance feedback. The basic idea of relevance feedback is to improve search performance for a particular query by modifying the query step by step, based on the user's judgment of the relevance or irrelevance of some of the images retrieved in the initial query. The relevant images are called positive examples and the irrelevant images are called negative examples.

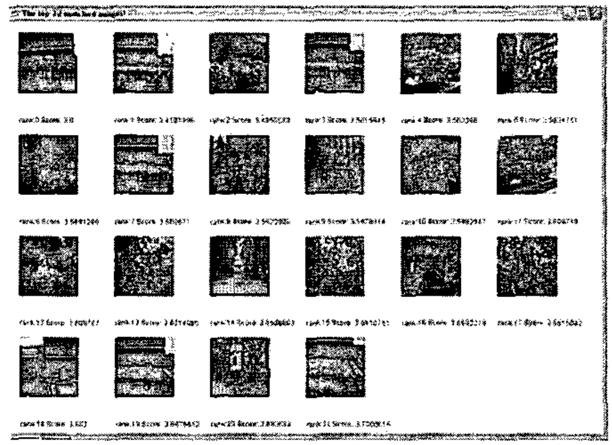

Figure 2. Image retrieval without Feedback

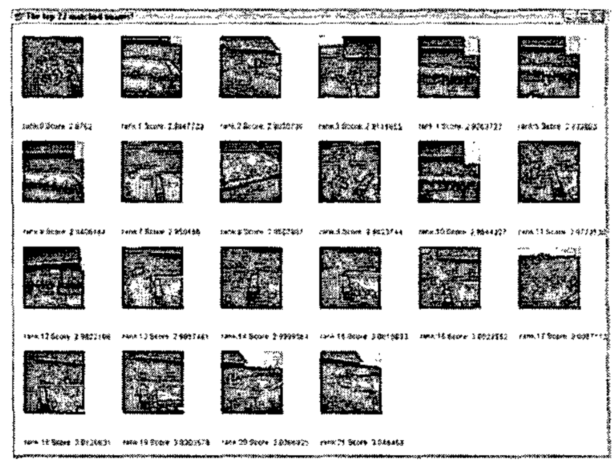

Figure 3. Image retrieval with feedback

After we obtain the positive and negative examples, relevance feedback can be realized by modifying the similarity measure. The similarity measure used for relevance feedback in DIRECT is

$$D(I_{Pos}^1, \ldots, I_{pos}^N, I_{Neg}^1, \ldots, I_{Neg}^M I_D)$$
$$= D(I_{pos}^1, \ldots, I_{pos}^N, I_D) \left[\frac{D(I_{pos}^1, \ldots, I_{pos}^N, I_D)}{D_N(I_{Neg}^1, \ldots, I_{Neg}^M, I_D)} \right]^K \qquad (6)$$

where I_{pos}^i and I_{Neg}^j are the i-th positive sample and j-th negative sample obtained from the retrieved images from previous retrieval, I_D is the image from the database. $D(I_{pos}^1, I_{pos}^2, \ldots, I_{pos}^N, I_D)$ is the similarity between the positive samples and image from the database,

$D(I_{Neg}^1, I_{Neg}^2, \ldots, I_{Neg}^M, I_D)$ is the similarity between the negative samples and image from the database. D is a similarity metric. For example, if we only use the color features, D can represent the result of multi-histogram intersection similarity measure developed in [Tang and Acton 2003].

Figure 2 shows a query result by only using color features and the similarity measure (3). Figure 3 displays the query results using multi-histogram intersection. From the example, we can see qualitatively that relevance feedback has improved the searching performance.

3 DECENTRALIZED PEER-TO-PEER FILE RETRIEVAL IN DIRECT

A major component of the DIRECT service is supporting decentralized peer-to-peer file retrieval. DIRECT will work as a service in the peer-to-peer environment of the NSDL collections to retrieve images from image collections. All participants in the system such as image collections and the DIRECT searching service will be regarded as peers, and control of the system will be distributed throughout these peers (see Figure 4) .

DIRECT creates a network of services by creating a brokered peer-to-peer system. A broker is an entity that connects service providers with service users or clients in a way that prevents either party from needing to know about the other *a priori* [Knighten 2000]. The service provider informs the broker that it provides a certain type of service. The client queries the broker for a certain type of service and, if there is a match, connects the client with the service provider. In the DIRECT approach, the portals are clients; the collections are service providers, and entities such as the index and the content based image retrieval portion of DIRECT, are both clients and service providers.

Currently, DIRECT adopts Java's Jini [Oaks 2000] to implement a broker in a peer-to-peer system. Jini provides an application called a lookup server that acts as a broker for the system. The lookup server comes with a predefined API (Application Program Interface) for finding and accessing it. When a service provider registers a service, it gives the Jini lookup server a set of attributes describing the service and a service object, which is usually a *stub* to a remote object. A stub is a special object that represents an object that is located in another process or on another machine. The stub has the same methods as the remote object it represents and when the program calls a method on the stub, the stub communicates across the wire to the remote object and calls that method on the remote object. A stub is a mechanism for allowing an object to appear local, when the object actually resides remotely [Oaks 2000].

When the client contacts the Jini lookup server, the client can search for the right service by the attributes given when the services were registered. When the client finds the appropriate service, the stub is sent, and then the client can call methods on the remote object through the stub.

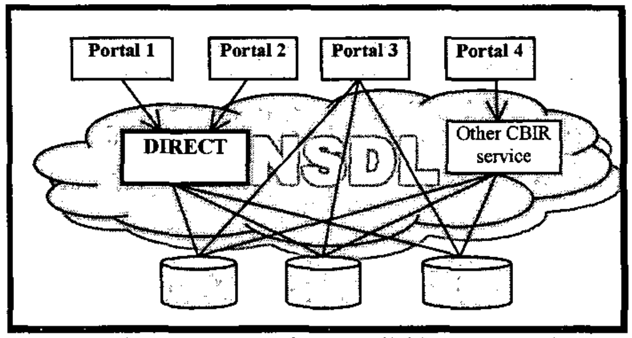

Figure 4. The DIRECT service is available to all portals. Here, Portals 1 and 2 use the service. Portal 3 does not and searches the collections directly. Portal 4 has implemented its own CBIR service. Both CBIR services search all collections.

The power of Jini resides in the "write once, run anywhere" idea of Java [Oaks 2000]. The stub object is written in Java, and the code for it can literally be sent across the wire to a remote machine and executed there, even if the machine has never seen the code previously. Therefore, a portal does not need to know how a collection implements a search protocol to query it. For instance, UC Berkeley's Simple Digital Library Interface Protocol [SDLIP 1999] defines a Java API but also defines several ways the stub can communicate with the remote object (e.g., Hypertext Transfer Protocol (HTTP) or Common Object Request Broker Architecture (CORBA)). This ability to allow the service provider to choose the form of communication is an important aspect of a decentralized service such as DIRECT.

The Jini lookup servers enable a decentralized system, but the broker itself is not only centralized but a single point of failure. To solve this dilemma, DIRECT executes multiple lookup servers in different geographic areas, and services is registered with each of the lookup servers. DIRECT allows a lookup server to fail and not debilitate the entire library, and also prevents the broker mechanism from being centralized with one system.

Another feature of Jini is the event system. The Jini lookup server will broadcast an event whenever a new service is added, or if an existing service is removed or modified. Clients can register to receive these notifications. The indexes use this mechanism to keep up to date. They receive notification whenever a collection joins, leaves, or announces its metadata has changed. The index then queries the collection to update its tables. The event mechanism allows a client to react to a service provider without having the service provider know about the client.

Jini provides the functionality to implement DIRECT as a decentralized peer-to-peer service. Jini supplies a broker so that DIRECT can find collections dynamically and allows mobile software that can communicate with each collection using the collection's desired communication mechanism. Last, Jini provides an event mechanism so that DIRECT can be notified when it needs to update its index.

4 CONCLUSION

We have described DIRECT, a decentralized image retrieval system for the NSDL. DIRECT uses distributed database technology to exchange digital images in a peer-to-peer environment. With a flexible, sustainable decentralized approach, the existing NSDL collections and services do not have to be significantly modified to utilize DIRECT. Most importantly, DIRECT gives the ability to search NSDL images by content. Such a content based image retrieval tool will allow access to the library by underrepresented groups such as non-English speakers. The service will also allow search and retrieval for images that have not been annotated manually. As the number of digital images in the NSDL explodes, automatic description of the digital image content, as proposed with DIRECT, becomes a critical need.

ACKNOWLEDGMENTS

The authors wish to thank Rob Antonucci of SSAI for help with the description of the distributed system. This work was supported in part by the NSF (0121596).

REFERENCES

Bach, J.R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R.C. and Shu, C. 1996 . Virage image search engine: an open framework for image management. In *Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV*, 2670, 76–87.

Bovik, A.C., Clark, M. and Geisler, W.S. 1990. Multichannel texture analysis using localized spatial filters. *IEEE Trans. PAMI*, 12, 55-73.

Do, M. and Vetterli, M. 2002. Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Transaction on Image Processing*, 11, 146-158.

Flickner, M., Sawhney, H., Niblack, W. and Ashley, J. 1995. Query by image and video content: The QBIC system. *IEEE Computer*, 28, 23-33.

Haralick, R. and Bosley, R. 1973. Texture features for image classification. In *Third ERTS Symposium*, NASA SP 351, Washington, D.C., 1219-1228.

Knighten, B. 2000. Peer-to-Peer Computing. http://cnscenter.future.co.kr/resource/hot-topic/p2p/PtP_IDF_Rev1.11-web.pdf

Ma, Y. and Manjunath, B.S. 1997. NETRA: A toolbox for navigating large image databases. In *IEEE International Conference on Image Processing*.

Mallat, S. 1989. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans. Pattern Anal. Machine Intel.*, 11,7, 674-693.

NSF-02-054. 2002. National Science, Technology, Engineering, and Mathematics Education Digital Library (NSDL)--- program Solicitation, http://www.nsf.gov/pubsys/ods/getpub.cfm?nsf02054.

Oaks, S., Wong, H. 2000. *Jini in a Nutshell*. O'Reilly & Assoc., Inc. Cambridge.

Pentland, A., Picard, R.W., and Sclaroff, S. 1996. Photobook: Content-based Manipulation of Image Databases. *Int. Journal of Computer Vision*, 18, 3, 233-254.

SDLIP. 1999. The Simple Digital Library Interoperability Protocol. http://www.dglib.stanford.edu/~testbed/doc2/SDLIP/SDLIPProjDesc/sdlipProject.htm.

Smith, J.R. and Chang, S.F. 1996. Querying by Color Regions using the VisualSEEk Content-Based Visual Query System. In *Intelligent Multimedia Information Retrieval*, Ed. Mark T. Maybury.

Swain, M.and Ballard, D. 1991. Colour Indexing. *International Journal of Computer Vision*, 7, 1,11-32.

Tang, J. and Acton, S. 2003. An Image Retrieval Algorithm Using multiple query Images. Submitted to *Seventh International Symposium on Signal Processing and its Applications*, July 1-4, 2003, Paris, France.

AUTHOR BIOGRAPHIES

JINSHAN TANG received his Ph.D. degree from Beijing University of Posts and Telecommunications, P.R. China, in 1998. From September 1998 to June 2000, he worked as an invited researcher in ATR Media Integration and Communications Research Laboratories (MIC), Kyoto, Japan. In June 2000, he went to Harvard to work on image enhancement technology. Since Dec. 2001, Dr. Tang has been working as a research scientist in Department of Electrical and Computer Engineering, University of Virginia. Dr. Tang is a senior member of the IEEE. His research interests are smart computers, content based image retrieval, cell detection and tracking, face detection and recognition, image enhancement for low-vision.

SCOTT T. ACTON received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Texas at Austin in 1990 and 1993, respectively. He has a B.S. degree in Electrical Engineering from Virginia

Tech (1988). He has worked in industry for AT&T, the MITRE Corporation and Motorola, Inc. and in academia for Oklahoma State University. For his research in video tracking, Dr. Acton was given an ARO Young Investigator Award. He received the Halliburton Outstanding Young Faculty Award in 1998. In 1997, he was named the Eta Kappa Nu Outstanding Young Electrical Engineer -- a national award that has been given annually since 1936. Currently, Dr. Acton is the Walter N. Munster Associate Professor and Faculty Fellow in the Department of Electrical and Computer Engineering at the University of Virginia. At U.Va., he was named the Outstanding New Teacher in Electrical and Computer Engineering. Dr. Acton is a senior member of the IEEE and has served as Associate Editor for the *IEEE Transactions on Image Processing* and the *IEEE Signal Processing Letters*.