Web Image Retrieval Using Self-Organizing Feature Map

Qishi Wu, S. Sitharama Iyengar, and Mengxia Zhu
Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803;
E-mail: wuq@csc.lsu.edu; iyengar@csc.lsu.edu; mzh1@lsu.edu

The explosive growth of digital image collections on the Web sites is calling for an efficient and intelligent method of browsing, searching, and retrieving images. In this article, an artificial neural network (ANN)-based approach is proposed to explore a promising solution to the Web image retrieval (IR). Compared with other image retrieval methods, this new approach has the following characteristics. First of all, the Content-Based features have been combined with Text-Based features to improve retrieval performance. Instead of solely relying on low-level visual features and high-level concepts, we also take the textual features into consideration, which are automatically extracted from image names, alternative names, page titles, surrounding texts, URLs, etc. Secondly, the Kohonen neural network model is introduced and led into the image retrieval process. Due to its self-organizing property, the cognitive knowledge is learned, accumulated, and solidified during the unsupervised training process. The architecture is presented to illustrate the main conceptual components and mechanism of the proposed image retrieval system. To demonstrate the superiority of the new IR system over other IR systems, the retrieval result of a test example is also given in the article.

1. Introduction

In recent years, with the rapid development of computer technologies and broad applications of the World Wide Web, large amounts of digital data being stored, transmitted, and accessed through the Internet have been explosively growing in the format of image, graphics, text, video, audio, etc. As one of the critical functionalities in the Web search engine, information retrieval has been attracting more attention than ever. Among the above various media types, images are of principal importance not only for its enormous popularity but the fact that images are the main carrier of complex and colossal information on the Internet. Due to the difficulty in interpreting human perception subjectivity of image content, it is encouraging to take both visual and textual features into consideration for image retrieval. In this article, we choose the Kohonen neural network model as the integration scheme of multimedia and multimodalities because of its properties of unsupervised learning and self-organizing. As a matter of fact, the history of image retrieval is the history of emulating the way in which the human brain would do in image discern. The day when we can completely simulate the performance of our human brain is the day when we can get to the apex of image retrieval system.

This article is organized as follows: Section 2 provides an overview of current image retrieval systems. In Section 3, the architecture of the SOFM-based image retrieval system is presented with the focus on the discussion of each module. Section 4 introduces the Kohonen model, and describes the implementation details of SOFM computation and its learning algorithm. A test example is given in Section 5 to show the performance of the proposed new approach. Section 6 draws conclusions for the article.

2. Overview of Current IR Systems

Key Word-Based IR

The traditional image retrieval system can be traced back to 1970s and was text based. The bottom line of this framework is annotating images manually first and then using text-based Database Management Systems (DBMS) to conduct the image retrieval (Rui, Huang, & Chang, 1997). It performed well, and many advances were being made during that period. However, two main problems exist in this method. First of all, the vast amount of human labor is required in the annotation of images. As the size of image repositories increases tremendously, this problem becomes more and more acute. Second, there is difference in the subjectivity of human perception. Different people would annotate the same image differently according to their own perception understanding. The mismatch between image annotation and query expressions would probably result in vain retrieval.

Content-Based IR

Another method, which was called as content-based image retrieval, was proposed in the early 1990s aimed to

overcome the above difficulties. Instead of manually annotating the image, images were indexed by their own low-level features, such as color, texture, shape, etc. A lot of work has been done on CBIR systems as to the feature extraction, multidimensional reduction, and indexing as well as matching techniques (Do, 1998; Pecenovic, 1997; Zachary & Iyengar, 2001).

On the other hand, despite these inspiring achievements, irrelevant images, especially when dealing with heterogeneous image collections from Web sites, is still a plague. Generally speaking, there is no direct link between low-level features and high-level features. The semantic gap is the inborn incapability of CBIR. For example, CBIR would sense a bunch of apples to be similar to a bunch of tomatoes. On the other hand, CBIR would judge that a sleeping white cat is very different from a standing black cat, based on color features or shape features. CBIR cannot perceive the most apparent semantic content from the image as can be easily done by the human brain.

Relevance Feedback for CBIR

To exploit user's interactive feedback, relevance feedback technique in content-based image retrieval was proposed (Rui, Huang, & Ortega, 1998). After getting the first set of results, the user is asked to give a quick view on the results and submit the feedback to dynamically update the different features' weight and refine the query to mirror the user's semantic query and subjective perception. This model is demonstrated to be somewhat intelligent because of the involvement of user's interactions. There are still some problems in this model due to the computational complexity, which would probably result in exceeding the time constraint, inflexibility in the model expansion, and hard system maintenance. Furthermore, this model does not support multiple query image vectors, which is always the case in real life. There are always several typical image vectors corresponding to a certain query; the only difference might be the statistical distribution probability.

3. SOFM-Based Image Retrieval System

Architecture of SOFM-Based Image Retrieval System

A general architecture of SOFM-based image retrieval system is shown in Figure 1. Basically, it consists of three main components: user interface, SOFM sample training, and image matching as well as image collection and feature extraction. The user interface is an interactive screen, which allows users to communicate with the system. The SOFM model is an intelligent unit serving as the brain of the IR system. Image retrieval is done both on-line and off-line. With the idea to meet the real time requirement, we need to put work as much as possible in the off-line part. The training of SOFM is a significant part of off-line computation. The weight matrix for each object category is stored in the parameter database as a training result. The World Wide Web is the ultimate data source for images to be retrieved. Source images reposit in the image and feature database after having been processed by feature extractor and indexer to speed up the image matching.

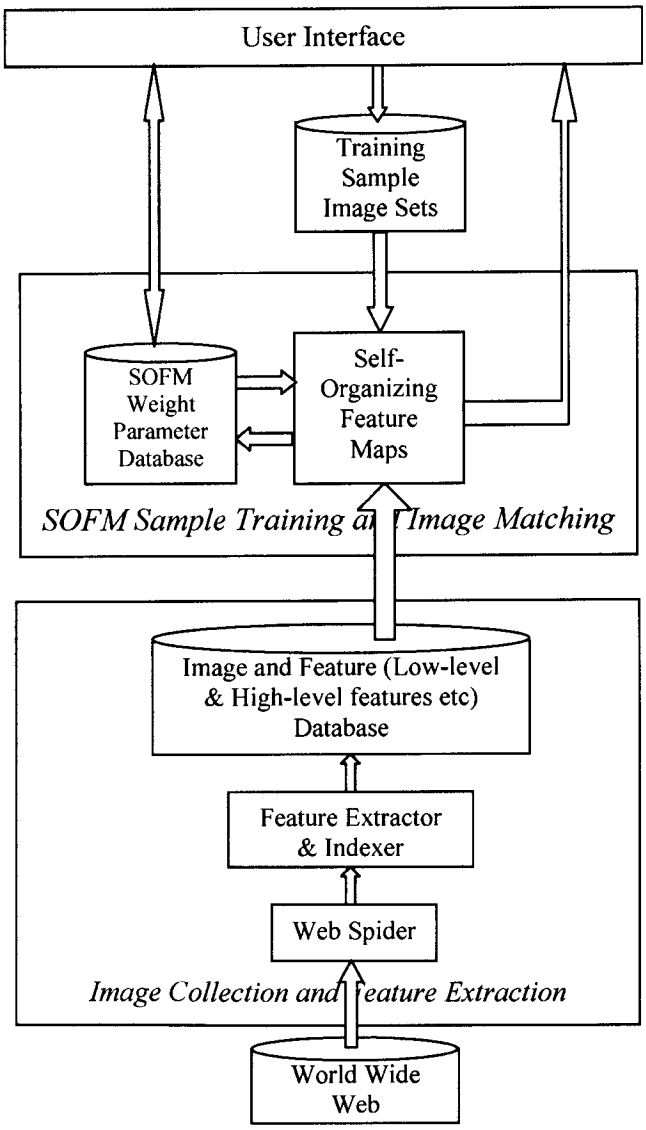

FIG. 1. Architecture of SOFM-based Web Image Retrieval System.

Image Collection and Feature Extraction

Image collection. Image collection is the first step that needs to be taken for image retrieval. There are immense amounts of images distributed on the whole Internet. Collecting candidate images is made possible by a Web spider, which is a kind of mobile agent. Mobile agents are programs that can be dispatched from one computer and transported to a remote computer for execution. While arriving at the remote site, they present their credentials and obtain access to local services and data to collect needed information or perform some certain actions and then return with results. Specifically, the Web spider is able to travel along various

designated Web pages as well as their hyperlinks to analyze and download interesting images to a local Web image database. This may be done intermittently.

Feature extraction. Feature extraction is a critical and indispensable component of image retrieval, which prepares the input vector for the upper SOFM model. Because of the heart position of SOFM model in the system, the qualities and types of feature extraction have significant impact on the performance of the IR system. We discuss the extraction of textural features and four types of low-level visual features in our article: color histogram, entropy, shape, and texture.

Textual feature. High-level textual features always play a crucial role in image retrieval. In our system, the semantic features of the image will be automatically extracted by way of keyword matching we describe below. The image-related texts involve different categories such as filename, alternative name, surrounding contents, URL, page title, etc., which are collected along with the image from the Web site by the Web spider. The textual feature similarity is computed from the match degree between m query keywords and n categories of retrieved texts.

$$MD = \sum_{i=1}^{m} \sum_{j=1}^{n} (\varepsilon_{ij} \cdot m_{ij})$$

where, ε_{ij} is the coefficient associated with query keyword k_i and category text s_j, and m_{ij} is their corresponding matching rate. Normalization may be favorable for better performance before *MD* is delivered to the SOFM model as one element of the input vector.

An alternative way to compute the textual feature similarity is to explore the pseudo keywords beforehand among those image-related texts according to their occurrence frequency. Suppose we consider n categories of retrieved texts, and use n character strings $s_1, s_2, \ldots, s_n$ to represent the text of each category. We find the first-order keywords by computing the LCS (longest Common Subsequence) among m strings.

$$s_1, s_2, \ldots, s_n \Rightarrow \text{LCS} \Rightarrow \text{the first order pseudo keywords.}$$

The second-order pseudo keywords is computed from $\binom{n}{n-1}$ string groups, each of which $S_i, S_{i+1}, \ldots, s_j$ has $n - 1$ strings,

$$s_i, s_{i+1}, \ldots, s_j \Rightarrow \text{LCS} \Rightarrow \text{the second-order pseudo keywords,}$$

and so on.

The query keywords will be compared with the above pseudo keywords to determine the textual feature similarity between candidate image and requested image. This method substitutes the manual annotation work to some extent. However, it requires more computation than the first method.

Color histograms. Color features are invariant to rotation, shift, and scaling, which motivate us to use it as a key feature in our system. The lack of a luminance channel and correlation in channels in the traditional RGB color space forced us to switch to CIE XYZ color space (Pecenovic, 1997). As one of the alternatives to RGB color spaces, CIELAB color space, first introduced in 1976, is frequently used to specify a three-dimensional color feature vector. In the CIELAB color space, L represents the brightness of the color, A and B are defined by an opponent color theory, in which A describes the redness to greenness, and B describes the yellowness to blueness of the color. Three channels in CIELAB color space are less correlated. Due to its three properties of uniformity, completeness, and uniqueness, CIELAB color space similarity is in better accordance with human perceptual similarity compared with popular RGB color (Zachary & Iyengar, 2001). We apply the CIELAB color space in our article. The color space is represented by color histogram, which reveals the distribution of color components in an image. Figure 2 shows two different images and their LAB components, respectively.

Image entropy feature. Color histograms serve as a good component of image feature vector. However, for very large image databases and histogram spaces with large dimensions, the computational cost would be considerably high. Image entropy, a measure of the complexity of image color distribution, maps an n-dimensional vector to the set of real number. Given a vector v of numbers from a set $\{x_1, x_2, \ldots, x_n\}$ where the probability that $x_i \in v$ is $p_i = P(x_i)$, the entropy of v is given by the formula:

$$H(v) = -\sum_{i=1}^{M} p_i \log(p_i)$$

According to the above mathematical description of entropy, images with simple color distribution have low entropy value, while images with complex color distribution have high entropy value (Zachary & Iyengar, 2001).

Shape feature. Shape is another important visual feature type helping people recognize images. Edge detection is a fundamental technique of image processing to obtain and utilize shape features. Figures 3 and 4 each shows an image contour extracted from a building and a bird image respectively, using a least-square–based edge point detection algorithm.

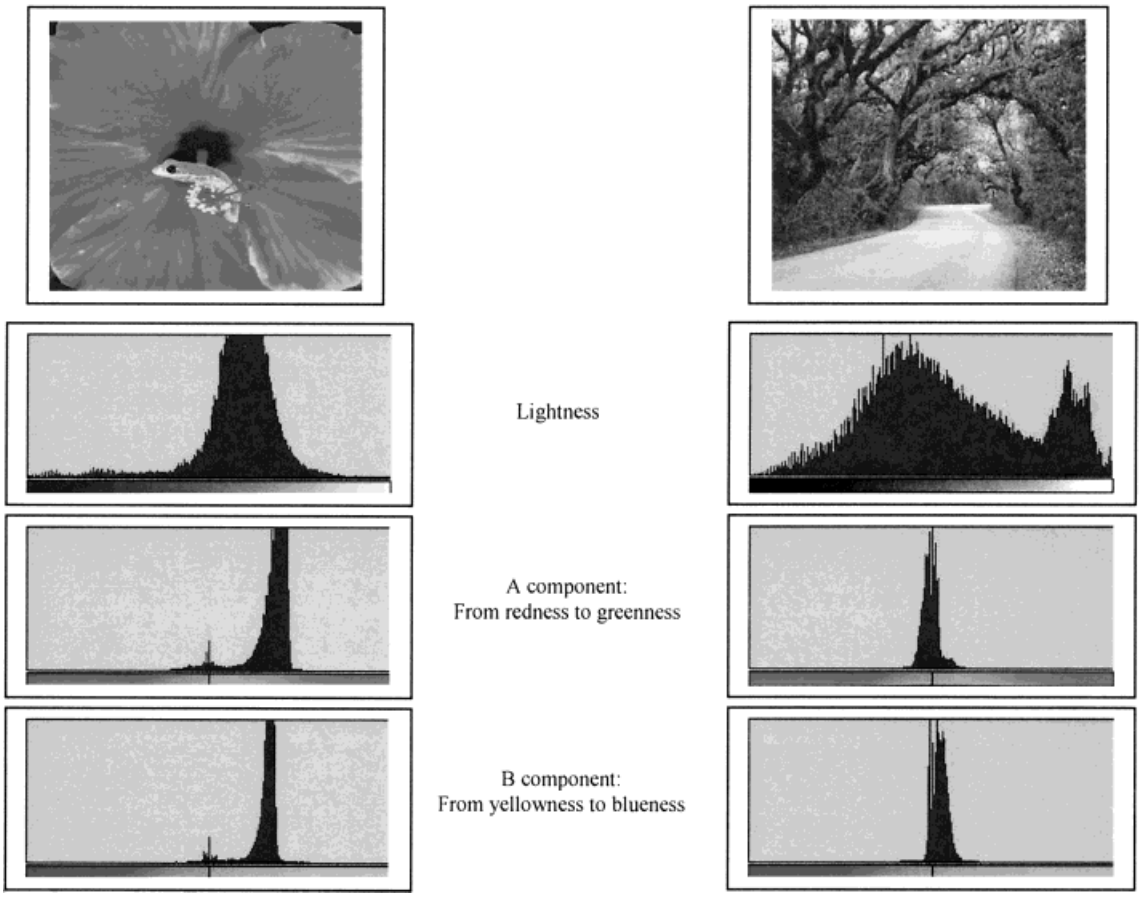

FIG. 2. A red flower image and a tree image and histograms of their LAB components.

To better assist in recognition of image contents, the result images from edge detection may be further processed to acquire a vectorized format by forming up edge points into typical shapes such as lines, polygons, ellipses, circles, etc. This is particularly favorable for recognizing those images with characteristic shapes. As Figure 5 shows, a characteristic rectangle is brought out from the edge points extracted from a door image by Hough Transform.

It should be indicated that for images with too complex a shape or without any characteristic shapes, shape extraction becomes extremely difficult to carry out. In the worst case, the extraction results may turn out to be completely useless. Image segmentation is usually applied to get the most significant part of the image for shape extraction and matching.

Texture feature. Texture features generally have richer information than color histograms and correspond to human perception rather well. The significant weakness is that

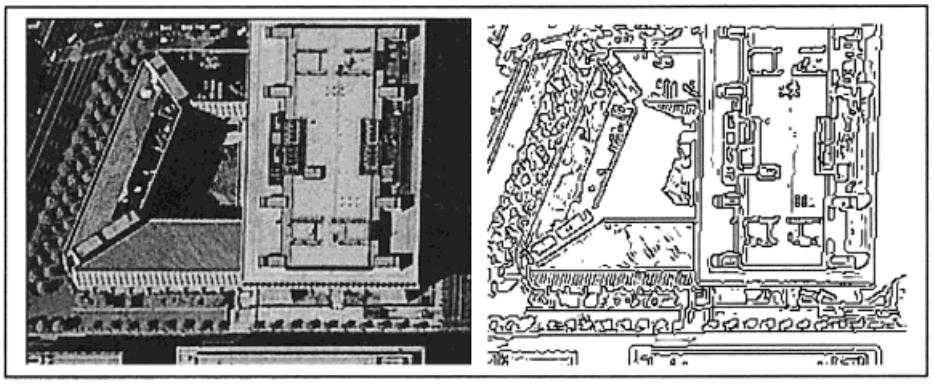

FIG. 3. Edge detection for a building image using a least-square–based algorithm.

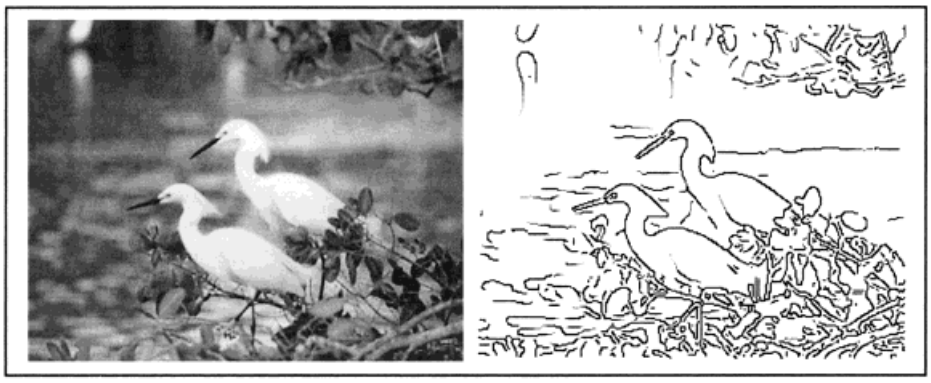

FIG. 4. Edge detection for a bird image using a least-square–based algorithm.

transforms used to extract texture features such as scaling, illumination, and view-angle is very sensitive (Pecenovic, 1997). Texture features can be represented by directionality, periodicity, randomness, and so on.

SOFM Sample Training and Image Matching

Constructing a high-quality SOFM is the key to the success of the IR system. It is a crucial part of work to select an appropriate SOFM model and prepare training sample image sets to train SOFM for each object category. The training of SOFM is essentially an unsupervised learning process. That means SOFM is able to apprehend and memorize the common characteristics of training samples just like our brain system. A new SOFM model always starts with initial random weight values, and ends up with a steady and convergent system after a certain number of learning cycles. Once the training process is done, the SOFM model as well as its associated weight matrix is stored in the parameter database for the object category being trained. Some characteristic weight vectors are always calculated from the vector clusters of weight matrix.

A corresponding SOFM model and its associated weight matrix is procured from the parameter database upon user's input of an image query. The distance between the normalized feature vector of an indexed image and the characteristic weight vector determines the similarity of the candidate image and requested object. In other words, the image similarity is defined as the distance or as the angel between two normalized vectors in the n-dimensional space.

$$\text{Dist} = \|V_f - V_w\| \quad \text{or} \quad \text{Ang} = \frac{V_f \cdot V_w}{\|V_f\| \, \|V_w\|}$$

where V_f is the feature vector extracted from a candidate image, and V_w is the characteristic weight vector of the image object to be retrieved.

In our IR system, the similarity is also demonstrated by the activation of neurons in the feature map. The detailed algorithm implementation will be discussed in Section 4.

A General Comparison of Various IR Systems

To show the superiority of the proposed IR system, Table 1 is provided below for comparison between different IR systems.

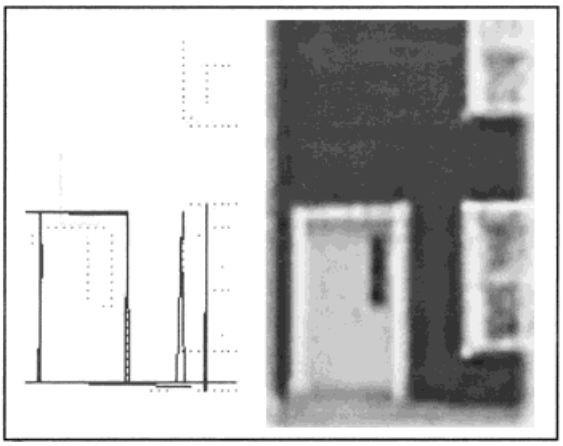

FIG. 5. A door image and its vectorized rectangular shape feature.

4. Implementation of SOFM Module

Brief Introduction to SOFM

Evolving from neuron-biological system, artificial neural network technology gives computers an amazing capacity to actually learn from input data and provides solution to problems, which usually demand human-like intelligence. The Kohonen Self-Organizing Feature Map, first introduced by Finnish professor Teuvo Kohonen (University of Helsinki) in 1982, is probably one of the most promising artificial neural network models, with aspect to emulating the learning process of the human brain. It is well known that the cortex of the human brain is subdivided in different regions, and each of them is responsible for certain functions. The neurons group themselves together in certain regions of cortex and each group responds to certain incoming information. The brain is a self-learning system in which

TABLE 1. Comparison between various image retrieval systems.

IR schemes \ Characteristics	Huge amount of human labor?	User's feedback required?	Vast amount of computational complexity?	Easy to expand or maintain the framework?	Multiple reference feature vectors allowed?	Mimic human brain mapping feature most naturally?
Keyword Based IR	Y	N	N	Y	N	N
Content-Based IR	N	N	Y	N	N	N
Relevance feedback IR	N	Y	Y	N	N	N
SOFM based IR	N	N	N	Y	Y	Y

the interconnections (synapse) between neurons can be changed. This unsupervised learning concept in the human zbrain is embedded into the Kohonen algorithm in a simplified way. In an SOFM model, neurons are able to organize themselves spontaneously to a certain pattern on a feature map according to certain input values.

The Kohonen learning algorithm is a competitive learning process, which means that neurons would compete for the privilege of learning, and no target pattern is given. The only winner is the neuron with maximum dot product of the current normalized input vector and its weight vector. Only the winning neuron and its neighborhood neurons are allowed to learn (adjust weight vectors) at certain rate. After an adequate learning process, the neuron weight map would reflect the distribution pattern of input vectors. SOFM is usually a two-layer network, but because vector normalization is required, there may exist one additional normalization layer, which ensures that all vectors lie within bounds. There are several normalization methods, of which simple length adjustment and Z-axis normalization are frequently used.

Two-Layer SOFM Model

Generally, there are two layers existing in the SOFM model: input layer and feature map layer, as Figure 6 shows.

The Input Layer takes multidimensional input patterns from external environment. In our IR system, an image is represented by an affiliated input vector, which may consist of textual features and visual features such as color, texture, shape, etc. All the features are extracted from that image. The number of neurons in the input layer is determined by the number of dimensions of the input vector. In Figure 5 there are two input neurons.

The Feature Map Layer is made up of M∗N neurons with associated weight vectors (Fig. 6 shows a 3∗3 feature map). Initially, all the weight vectors are assigned random values and distributed randomly on the unit circle, as Figure 7 shows. Each neuron receives a sum of weighted input from the input layer and feature map layer. Neurons on the feature map layer are connected with some other neurons on the same layer, which make up its neighborhood. After receiving a given input, some neurons on the feature map layer would be activated. These activated neurons and their neighborhood neurons are allowed to modify their weight vectors at different levels according to the distance to the winning neuron. The change in the weight vectors would push the weight vectors to the input vector. After a successful learning process, the dominant part of weight vector clusters in the region with a high probability of input vectors and fewer is gathered in the region with a low probability of input vectors, or even no vector appears in the region without any input pattern, as Figure 8 shows. In other words, the neurons in the feature map layer would mirror the probability distribution of input images from the environment.

It is reasonable to observe that the feature patterns of the same object may map into more than one region on the

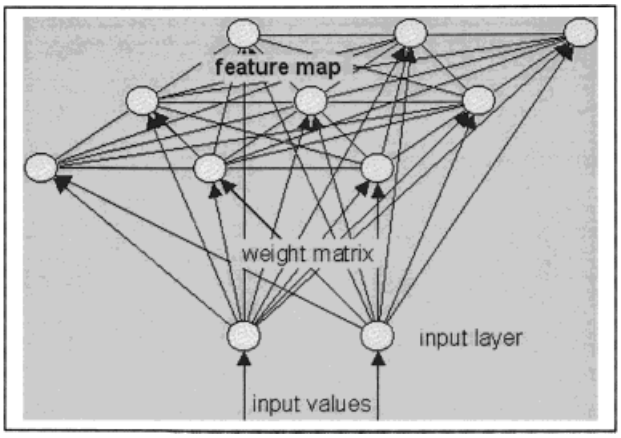

FIG. 6. Two layer self-organizing feature map model.

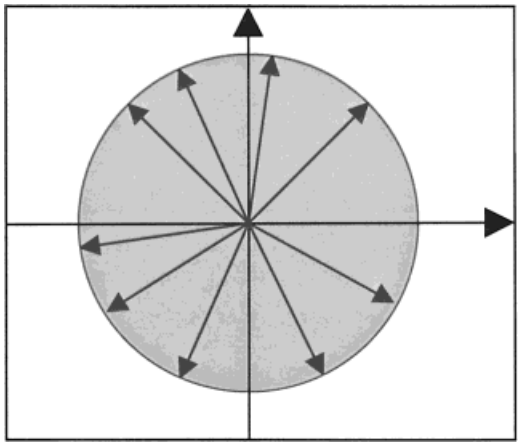

FIG. 7. Initial weight vectors distributed on the unit circle randomly.

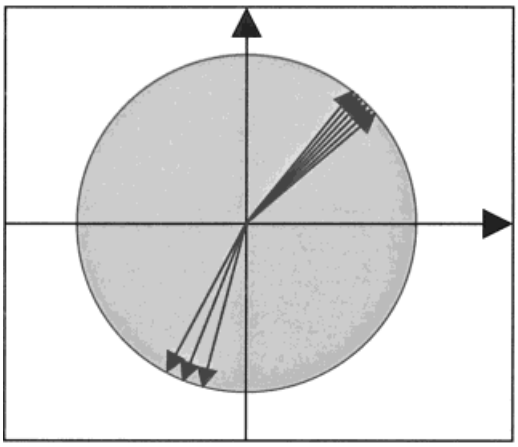

FIG. 8. Weight vectors cluster in certain regions after learning.

feature map. For example, a set of typical tree images could have two totally different feature patterns. Trees turn out to be green, and have flourish shapes in the spring and summer, while trees are always characterized by yellow and withered shapes in the fall and winter. It is a superior aspect of the SOFM-based IR system. In some other IR systems using a single-query image as a reference vector, yellow trees would be screened out if green trees were used as a query image, and vice versa. The SOFM-based IR system allows multiple query feature vectors, that is, it has a wide recognition tolerance just like the human brain system.

Kohonen Learning Algorithm

Kohonen neural network model has a relatively simple learning algorithm, which is basically an iterative computation process of making adjustment to the weight matrix. The objective of the learning algorithm for the SOFM neural network is to form a feature map, which captures the essential characteristics of the input vectors and maps them onto a typically 1D or 2D feature space (Vesanto & Alhoniemi, 2000). Unlike the BPN algorithm, the SOFM algorithm adopts an unsupervised learning technique and requires no target patterns for training sample inputs. The following depicts the main steps taken in the SOFM learning process in our system.

Step 1. System initialization. First of all, an SOFM model with appropriate dimensions of input layer and feature map is constructed according to the specific image object to be retrieved. Then we connect the input layer with the feature map by assigning random values for the weight matrix. The initial learning rate and activation area are also carefully specified as two important system parameters for fast learning speed and good learning performance.

Although it is commonly acceptable to randomly select a set of weights for the neurons, we may want to initialize the neuron weights to mirror the image inputs if we can get some ideas from the possible input image vectors.

Step 2. Determining the winning neuron for each input vector. The input vector of the training image is represented by $V(v_1, \ldots, v_n)$ and the neuron weight vector is represented by $W(w_{i1}, \ldots, w_{in})$, $i = 1, 2, \ldots, m$. Here, m is the number of neurons in the feature map. The winning neuron has the maximum value of the weighted sum.

$$\text{Max}\left\{S_i = \sum_{j=1}^{n} V_j \cdot W_{i,j}, \quad i = 1, 2, \ldots, m\right\}$$

Geometrically, the weighted sum is simply a dot product of the input image vector and the neuron weight vector.

$$W \cdot V = W_1 \cdot V_1 + \ldots + W_n \cdot V_n$$

Step 3. Calculating the neighborhood function. We calculate the neighborhood function as follows:

$$\Lambda_i = \exp\left(-\frac{d_i^2}{2\sigma^2}\right), \quad i = 1, 2, \ldots, m$$

where, $d_i = \|W_i - W_w\|$, $i = 1, 2, \ldots, m$ is the distance between winning neuron weight vector and all its neighborhood neurons. σ^2 is the variance parameter specifying the spread of the Gaussian function, and determines the neighborhood (activation area).

Step 4. Updating weight vectors for neighborhood neurons. Once the neighborhood function is obtained, all neurons in the winning neuron's neighbood area will have their weights adjusted by a strength proportional to the neighborhood function and to the distance of their weight vector from the current input vector.

$$\Delta W_i = \eta \cdot \Lambda_i \cdot (V - W_i),$$

$$W_{(\text{new})i} = W_{(\text{old})i} + \Delta W_i, \quad i = 1, 2, \ldots, m$$

where, η is the current learning rate.

Step 5. Adjusting system parameters. As the learning proceeds, the neighborhood (activation area) shrank until it included only one neuron, and the learning process is slowed down by reducing σ and η, respectively.

$$\sigma_{(\text{new})} = \sigma_{(\text{old})} \cdot \delta\sigma$$

$$\eta_{(\text{new})} = \eta_{(\text{old})} \cdot \delta\eta$$

5. Results of Testing Example

The proposed the SOFM model is trained by a series of simulated data. The testing images are collected from a few

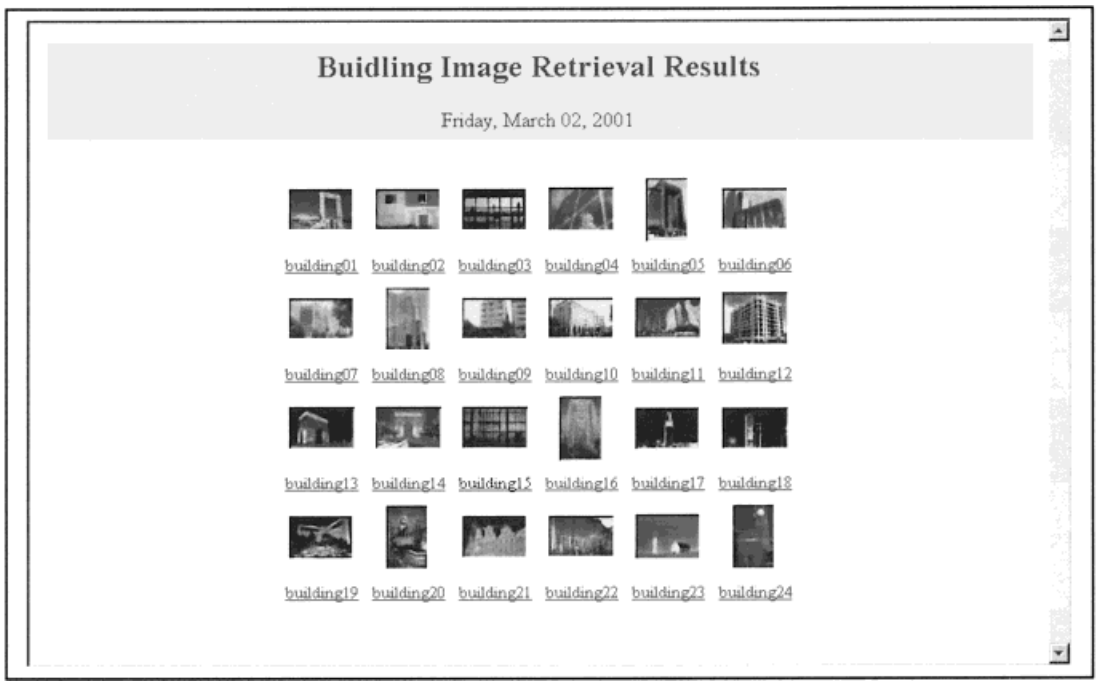

FIG. 9. Testing results for building image retrieval.

on-line and off-line image libraries. Those visual and textual features discussed above are extracted from testing images to form the representative image vectors. Figure 9 shows the testing results for architectural building image retrieval. According to the results, it has been observed that the shape features of building influentially contribute to the building image retrieval, and the component of color histogram in the feature vectors makes it possible to discriminate between evening scenes and daytime scenes.

6. Discussion and Conclusion

In this article, we propose a SOFM-based image retrieval system. The new system utilizes not only low-level visual features, but also high-level textual features. It has been shown that the combination of Content-based IR and Keyword-based IR can result in satisfactory retrieval performance. Furthermore, the new system allows multiple reference feature vectors by training the system with a set of sample images, which gives our system more flexibility. The system is also designed to be expandable with low computational expense. To ensure the system performance, selection of feature types is crucial. For different types of images, the selected feature types might be quite different. At this point, we have not taken the significance of feature types into consideration. If the weights of different feature types are dynamically assigned and adjusted during the training process, the retrieval performance would expect to be dramatically increased (Laaksonen, Koskela, Laakso, & Oja, 2000).

References

Chen, Z., & Wengyin, L. (1997). Web mining for web image retrieval. Microsoft Research China.

Do, M.N. (1998). Invariant image retrieval using wavelet maxima moment. Lausance: Swiss Federal Institute of Technology; SSC Doctoral School Project Report.

Laaksonen, J.T., Koskela, J.M., Laakso, S.P., & Oja, E. (2000). PicSOM—Content-based image retrieval with self-organizing maps. Pattern Recognition letters, 21 (13–14), 1199–1207.

Pecenovic, Z. (1997). Image retrieval using latent semantic indexing. Lausanne: Swiss Federal Institute of Technology, graduate thesis.

Rui, Y., Huang, T.S., & Chang, S.-F. (1997). Image retrieval: Past, present, and future. Proc. of Int. Symposium on Multimedia Information Processing.

Rui, Y., Huang, T.S., & Ortega, M. (1998). Relevance Feedback: A powerful Tool for Interactive Content-Based Image Retrieval, IEEE Transaction on Circuits and Video Technology.

Vesanto, J., & Alhoniemi, E. (2000). Clustering of the self-organizing map. IEEE Transactions on Neural Networks, 11 (3), 586–600.

Zachary, J., & Iyengar, S.S. (In press). On the use of information theory for computing similarity in content based image retrieval, 2001 International Conference on Imaging Science, Systems, and Technology, submitted.