PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 38 (2005) 707–722

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Content-based image retrieval using growing hierarchical self-organizing quadtree map

Sitao Wu, M.K.M. Rahman, Tommy W.S. Chow*

Department of Electronic Engineering, City University of Hong Kong, Hong Kong

Received 17 February 2004; accepted 25 October 2004

Abstract

In this paper, a growing hierarchical self-organizing quadtree map (GHSOQM) is proposed and used for a content-based image retrieval (CBIR) system. The incorporation of GHSOQM in a CBIR system organizes images in a hierarchical structure. The retrieval time by GHSOQM is less than that by using direct image comparison using a flat structure. Furthermore, the ability of incremental learning enables GHSOQM to be a prospective neural-network-based approach for CBIR systems. We also propose feature matrices, image distance and relevance feedback for region-based images in the GHSOQM-based CBIR system. Experimental results strongly demonstrate the effectiveness of the proposed system.

Keywords: Content-based image retrieval; Growing hierarchical self-organizing quadtree map; Image distance; Relevance feedback

1. Introduction

The technology for computing and storage has been rapidly evolved that people can collect and store information from a wide range of sources at rates that were unprecedentedly experienced a few years before. Digital images and videos are becoming more commonly used everywhere while the production of such multimedia information is growing at an unbelievable rate. To access, retrieve and browsing the large-scale digital images have become a very challenging and important task.

Content-based image retrieval (CBIR) is one of the most effective techniques for retrieving semantically relevant images from unlabelled image data sets based on automatically extracted features. It has been an ongoing research subject for more than a decade [1]. It usually retrieves relevant images based on the image comparison of visual contents, such as color, texture, shape, structure, etc. One of the problems associated with CBIR is that there is not an accepted standard criterion for judgement of relevancy of retrieved images. Human subjectivity is indispensable for the evaluation of CBIR system. Therefore it is difficult to compare the retrieved results from different CBIR systems.

A few CBIR systems have been implemented in recent years. The well-known and is the query by image content (QBIC) [2] developed at IBM is one of the earliest well-known commercial system. Other commercial systems are IBM T. J. Watson [3], VIRAGE [4], NEC AMORE [5], Bell Laboratory WALRUS [6], etc. MIT photobook [7] is one of the earliest academic CBIR systems. Other academic CBIR systems developed in recent years are Berkeley Blobworld [8], Columbia VisualSEEK and WebSEEK [9], CMU Informedia [10], UCSB NeTra [11], UCSD [12], University of Maryland [13], Standford EMD [14] and WBIIS [15], PSU SIMPLIcity [16], etc.

The simplest form of storing images is to arrange images in a flat structure. When a new image is queried, it is

* Corresponding author. Tel.: +852 2788 7756; fax: +852 2788 7791.
E-mail address: eetchow@cityu.edu.hk (T.W.S. Chow).

0031-3203/$30.00
doi:10.1016/j.patcog.2004.10.005

compared with all images stored in a database and the returned images are sorted according to image distances. The computational complexity of direct query without feedback is $O(n)$, where n is the number of images in the database. When new images are added into the databases, no extra steps are required to handle with the flat structure. The image query time can be reduced if images are organized in a hierarchical tree structure rather than a flat structure. TS-SOM [17] is such a tree-structured vector quantization algorithm that uses self-organizing map (SOM) [18] at each of its hierarchical levels. With neurons arranged in a tree structure, TS-SOM is able to reduce the computational complexity of search to $O(\log N)$. PicSOM [19,20] is a neural-network-based CBIR system that utilizes TS-SOM. Images are organized hierarchically from coarse top levels to precise bottom levels in PicSOM. The searching time of PicSOM is faster than that by the systems using the flat image structure. A problem of TS-SOM is that the number of neurons at the nth level is 2^n for 1-D SOM or 4^n for 2-D SOM. When n is 6 or more for a 2-D SOM, the number of neurons at the bottom level will exceed 4096. As a result, substantial computational effort is required. Furthermore, at such bottom level, a considerable number of neurons are not associated with images and is therefore wasteful. Another problem of TS-SOM in PicSOM is that it is used in a static environment where all training images are collected before training. In most real world applications, more new images are required to be added into the old database and TS-SOM must retrain the network by using all old and new images together. Apparently, this poses a major problem for the incremental learning of TS-SOM, whilst the aforesaid direct query does not experience such a problem.

In this paper, we propose a region-based CBIR system using a growing hierarchical self-organizing quadtree map (GHSOQM). Each image in our proposed CBIR system is firstly segmented into several regions. Each region has similar features for colors and textures. Each image is thus represented by a region-based feature matrix. Different images may have different number of regions. As far as the authors are aware, there, hitherto, has not been a definition of feature matrices for neurons in neural networks. We propose that all neurons in GHSOQM have a fixed number of row vectors in feature matrices, which means that all neurons represent images with fixed number of regions in feature space. This enables us to deal with neurons in neural-network-based CBIR systems. We also propose a new criterion for image distance that can be applied to region-based representation of images. The similarity between images is measured by the distance criterion. The more similar the two images, the smaller the distance between them will be. GHSOQM organizes images in hierarchical levels like TS-SOM. Since SOM usually defines an elastic topology-preserving net stretched in the input space, high-dimensional images can be arranged in a 2-D grid at different precision level in GHSOQM. Images belong to neighboring neurons have similar semantic meanings in an SOM of the same level. One of the advantages of the proposed GHSOQM over TS-SOM is that the low-level neurons of GHSOQM are generated adaptively whilst those of TS-SOM are a priori predefined. Thus, dead or useless neurons at each of the hierarchical level are removed. This is significant because the storing space for neurons is saved and searching time can be significantly reduced. Like TS-SOM, the image retrieval time is reduced attributed to the hierarchical tree structure. The most important contribution of GHSOQM is that GHSOQM can incrementally learn new incoming images without retraining all images. This is practically essential for dealing with very large image database. Coupled with a relevance feedback technique, the proposed CBIR system can achieve better retrieving results. Experimental results demonstrate the effectiveness of the proposed CBIR system.

The content of this paper is organized as follows. In Section 2, CBIR system is briefly reviewed. In Section 3, SOM and its variants are introduced. The new algorithm of GHSOQM is proposed. We present our proposed region-based CBIR system using GHSOQM in Section 4. Experimental results are provided in Section 5. Finally, conclusion is drawn in Section 6.

2. Previous work on CBIR

2.1. Features of images

The basic rule of a CBIR system is to extract features of each image from its pixels and use them for comparing images. So the high-dimensional images are compressed and represented by simpler and low-dimensional features. However, features of an image should have a strong relationship with semantic meaning of the image. The feature extraction is far from an image compression algorithm. The low-level signatures should match the semantic high-level concepts perceived by humans to understand images as much as possible.

A CBIR system tries to find relevant images from a query images and sort the retrieved images. All those are done according to a minimum distance or maximum similarity measure in the feature space. How to extract features from images, which features are to be used and how these features are to be weighted leave rooms for different algorithms. In general, the feature extractions in a CBIR system have the following three types [16]: (1) histogram extraction; (2) color layout extraction; (3) region-based extraction.

Region-based extraction is better than histogram or color layout extraction for CBIR systems [16]. It first partitions images into several regions, which may represent some semantic objects in images. Then it extracts features from segmented regions. Region-based extraction adaptively segments images according to each image while color layout scheme does those without a priori knowledge of images. Some features other than colors, such as textures and shapes, can be also included in region-based extraction.

Region-based extraction can have a better retrieval result if image segmentations are consistent with the objects in images. The performance of region-based extraction is robust to image shifting, scaling, cropping, rotation [16].

2.2. Relevance feedback

The gap between the high-level semantic concepts by human and the low-level features by computers makes some retrieved images by CBIR system irrelevant to query images. Motivated by the feedback in control theory, relevance feedback (RF) is introduced to obtain better retrieval results [21]. The basic idea of RF is to feed back the users' judgments of relevancy of retrieved images to query images and modify query feature vectors through several iterative procedures. It is a supervised learning and users act as teachers in RF systems. Usually after several iterations, the returned images are better than those by one-shot retrieval.

Suppose z is a feature vector of query image, $X = \{x_1, \ldots, x_n\}$ is the set of returned relevant n images and $Y = \{y_1, \ldots, y_m\}$ is the set of returned non-relevant m images. The new query vector is modified by

$$z(t+1) = \alpha z(t) + \frac{\beta}{|X|} \sum_{x_i \in X} x_i - \frac{\gamma}{|Y|} \sum_{y_i \in Y} y_i, \tag{1}$$

where $|X|$ and $|Y|$ are the number of elements in X and Y sets, respectively, and α, β and γ are weights for current query image, relevant images and irrelevant images, respectively.

2.3. Image distance measure

For histogram extractions, the distance between two images can simply be Euclidean distance. For region-based extraction, different images may have different numbers of segmented regions. Hence, the total numbers of features extracted from different images are different in spite of the same number of features extracted from each region. Therefore Euclidean distance cannot be applied directly for comparing region-based images. Comparing images with region-based features does not have accepted methods now. One method was developed by Smith and Li [3], where each image is represented by a composite region template (CRT) matrix and the distance of two images is measured by the closeness of the two matrices. However, their method is not robust to image shifting, scaling and rotation [16].

Integrated region matching (IRM) [16] is another method to compare images with region-based features. It computes weighted region-wise distances for the distance of two images. For example, there are two images $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$, where x_i and y_i are region feature vectors for image X and Y, respectively. IRM first computes the region-wise Euclidean distances between regions of image X and Y and obtains a $n \times m$ distance matrix D, where the ijth element is $D_{ij} = \|x_i - y_j\|$. Then IRM computes a $n \times m$ weight matrix W that considers the region importance between two images. Finally, IRM calculates the distance Dis between the two images by just weighted sum of D

$$Dis(X, Y) = \sum_{i,j} w_{ij} D_{ij}, \tag{2}$$

where w_{ij} is the ith row and jth column element of the matrix W and satisfies $w_{ij} \in [0, 1]$ and $\sum_{i,j} w_{ij} = 1$. From Eq. (2), it can be seen that the distance measure uses soft assignment since all pairs of the distances between the two regions of two images are considered and weighted.

However, the computation of the weight matrix W in IRM is quite complex and need considerable computation to process it. If only the nearest region of image Y from a region of image X is considered, which is called hard assignment, the computation of the matrix W can be avoided. In the next section, the hard assignment is introduced in our proposed new distance measure instead of the soft assignment in IRM.

3. Related work on SOM and GHSOQM

3.1. Related work on SOM

Self-organizing map (SOM) consists of M neurons located on a regular low-dimensional grid (usually 1-D or 2-D). The lattice of the 2-D grid is either hexagonal or rectangle. The basic SOM algorithm is iterative. Each neuron i has a d-dimensional feature vector $w_i = [w_{i1}, \ldots, w_{id}]$. At each training step t, a sample data vector $x(t)$ is randomly chosen from a training set. Distances between $x(t)$ and all feature vectors are computed. The winning neuron, denoted by c, is the neuron with the feature vector closest to $x(t)$

$$c = \arg\min_i \|x(t) - w_i\|, \quad i \in \{1, \ldots, M\}. \tag{3}$$

A set of neighboring nodes of the winning node is denoted as N_c, which decreases its neighboring radius of the winning neuron with time. We define $h_{ic}(t)$ as the neighborhood kernel function around the winning neuron c at time t. The neighborhood kernel function is a non-increasing function of time t and of the distance of neuron i from the winning neuron c in the 2-D output space. The kernel can be taken as a Gaussian function $h_{ic}(t) = \exp(-\|Pos_i - Pos_c\|^2/2\sigma^2(t))$, $i \in N_c$, where Pos_i is the coordinates of neuron i on the output grid and $\sigma(t)$ is the kernel width.

The weight-updating rule in the sequential SOM algorithm can be written as

$$w_i(t+1) = \begin{cases} w_i(t) + \varepsilon(t) h_{ic}(t)(x(t) - w_i(t)), & \forall\, i \in N_c \\ w_i(t), & \text{otherwise} \end{cases} \tag{4}$$

Both the learning rate $\varepsilon(t)$ and kernel width $\sigma(t)$ decrease monotonically with time.

The main drawback of SOM is that one must predefine the map structure and the map size before the commencement of the training process. SOM is inherently limited by the fixed network structure. Usually, we must adopt trial tests to select an appropriate network structure and size. The essential factor of eliciting these drawbacks is that we predefine the data structure, which instead should be determined by the data themselves. Several improved SOM algorithms and SOM-related algorithms have been proposed in recent years to overcome the predefined structure. The neural networks in these algorithms dynamically increase their networks to suitable sizes. These algorithms grow neurons either in input space [22–25], or in both input space and output space [26–28]. Some other algorithms grow SOM with quadtree structure [29] were also proposed in [30,31].

Hierarchical SOMs are other variations of SOM. The basic idea of hierarchical SOM is to use multiple SOMs from top low-resolution to bottom high-resolution levels. TS-SOM is the first tree-structured hierarchical SOM. Self-organizing tree map (SOTM) [32] and self-organizing tree algorithm (SOTA) [33] use hierarchical tree structures and the training algorithms are like SOM. But the neurons in each level for both algorithms are only one-dimensional. Growing hierarchical SOM (GHSOM) [34] is a hierarchical SOM. GHSOM can grow neurons horizontally at each level or vertically for the whole tree structure. However, all these variations of SOM are not capable of incremental learning that can learn new data without retraining all data. We propose GHSOQM with the ability of incremental learning in the next subsection.

3.2. GHSOQM

GHSOQM is a hierarchical growing SOM using a quadtree structure. A neuron at higher level can generate its child SOM at lower level according to the number of inputs associated to it. This is like GHSOM where the network grows hierarchically under some condition. But GHSOQM does not grow neurons horizontally because we want to simplify the growing process and train the network faster. A neuron in GHSOQM may have four child neurons upon some conditions. Fig. 1a shows the structure of GHSOQM, which is very similar to the quadtree structure. Note that the number of neurons at each level of GHSOQM is adaptively determined while that of TS-SOM is predefined. And the structure of GHSOQM can be considered as an incomplete one of TS-SOM. If we look down from the top of hierarchy of GHSOQM and only use the leaf neurons without child SOMs, the final one-level map is like a one-level quadtree-like SOM [30,31] as shown in Fig. 1b corresponding to Fig. 1a. In an SOM at each level of GHSOQM, neighboring neurons have similar input data belonging to them. Usually the root level with one neuron is useless and we begin with the first level with four neurons.

The training data set is denoted as $X = \{x_1, \ldots, x_n\}$. The input data associated with neuron i at the nth level are denoted by $X_n(i)$. The feature vectors of the child neurons at the $(n+1)$th level from the mother neuron i at the nth level is denoted by $W^i_{n+1} = \{W^i_{n+1}(1), W^i_{n+1}(2), W^i_{n+1}(3), W^i_{n+1}(4)\}$. Then the GHSOQM algorithm is summarized as follows:

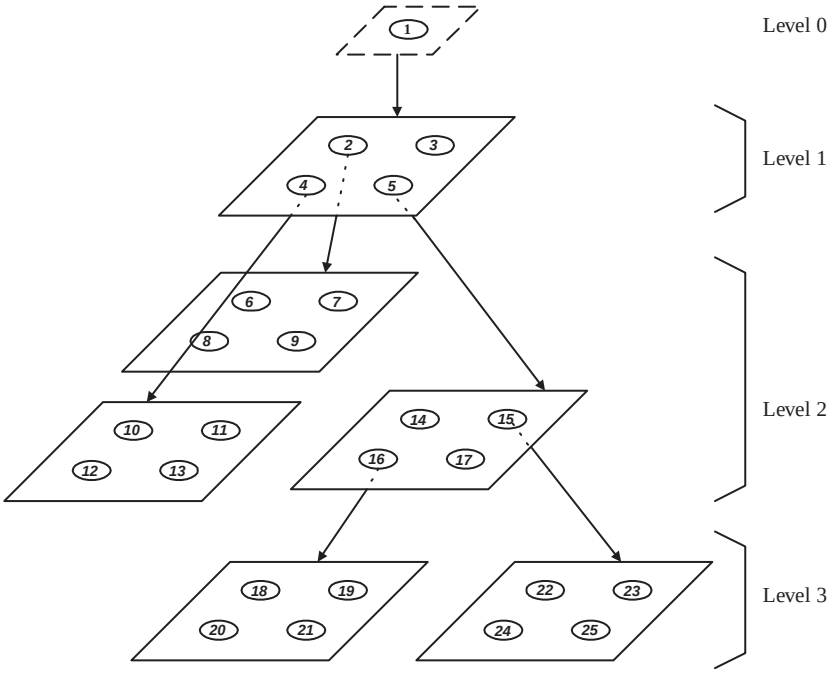
1. INITIALIZATION:
   Set level $n=1$ and the feature vectors at the first level $W^0_1 = \{W^0_1(1), W^0_1(2), W^0_1(3), W^0_1(4)\}$, where $W^0_1(i)$ is the feature vector of the ith neuron at level 1. An SOM with the four neurons is trained with all data X by invoking the function $TRAIN\_SOM(X, n, W^0_1)$.
2. RECURSIVE LOOP:
   $GENERATE\_SOM(X, n, W^0_1)$.

FUNCTION $GENERATE\_SOM(X, n, w)$
  FOR $i = 1$ to 4
    Assign each input datum in X to its nearest neurons. If the number of inputs assigned with the ith neuron at the nth level is more than a predefined number τ, then the neuron spawns four child neurons representing a child SOM with size 2×2. Then train the child SOM by $TRAIN\_SOM(X_n(i), \mathrm{n}+1, W^i_{n+1})$ and generate child SOMs by recursively invoking $GENERATE\_SOM(X_n(i), \mathrm{n}+1, W^i_{n+1})$.
  END

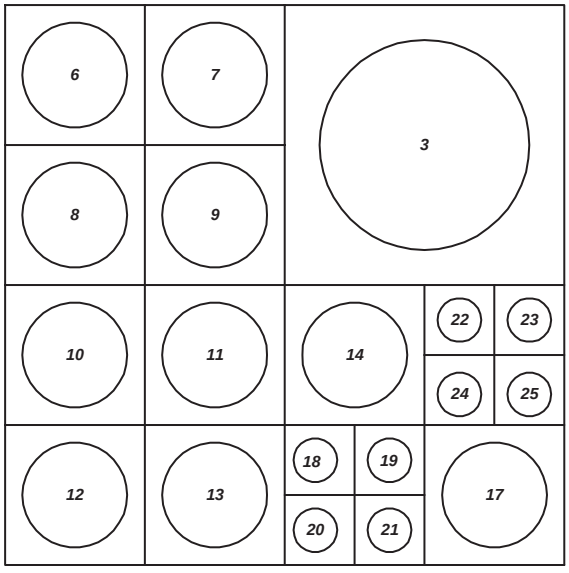
FUNCTION $TRAIN\_SOM(INPUT, n, W)$
  Train SOM at the nth level with the input data $INPUT$ and the four neurons with feature vectors W.

In the above GHSOQM algorithm, the determination of the value of τ depends on the number of input data. The more the number of input data, the larger the value of τ will be. Usually the value of τ can be taken as 1–5% of the number of input data for satisfactory performance.

The function *TRAIN_SOM* is an implementation of the original SOM algorithm. The function *GENERATE_SOM* recursively generates child SOMs if possible and train them with data associated with their mother neurons. In a word, GHSOQM trains SOMs at each level by the data associated with their mother neurons. While the training of all SOMs at all levels are completed simultaneously in the other hierarchical SOMs [32–34], GHSOQM completes the training of SOMs at the upper levels and then proceeds to train SOMs at the next lower level.



(a)



(b)

Fig. 1. (a) Architecture of GHSOQM that grows neurons hierarchically when needed. (b) The one-level map corresponding to (a) when we look down from the top of hierarchy of GHSOQM and only use the leaf neurons without child SOMs.

4. GHSOQM and relevance feedback in CBIR systems

4.1. Overall process of GHSOQM-based CBIR system

All images are processed by the same feature extraction method. Each image is first segmented into several similar regions by the JSEG algorithm [35], which can provide good segmentation results on a variety of images. The characteristic features, i.e., colors and textures, are extracted for each region of an image. After all available images are processed, GHSOQM is trained by using region-based feature vectors for images. After completion of training, all images

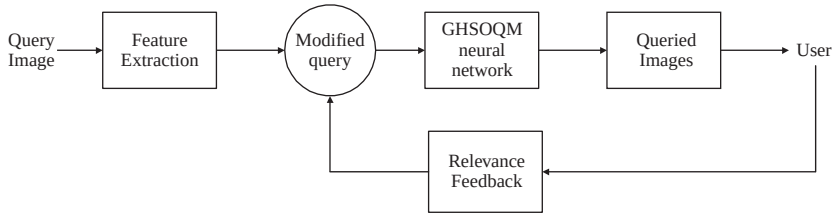


Fig. 2. Architecture of GHSOQM-based CBIR system.

are first assigned to the SOM at the first level according to the nearest distance. Then the images assigned to a neuron at the first level are assigned to the child neurons of the neuron. The assignment process proceeds until the leaf neurons are assigned with images. After completion of image assignment to neurons, the GHSOQM-based CBIR system is ready for query or retrieval. The image retrieval procedure can be described as the following steps:

Step 1: A submitted query image is processed to extract region-based features.

Step 2: The CBIR system first finds a nearest neuron at the top level of GHSOQM.

Step 3: If the number of associated images in the nearest neuron exceeds a prespecified minimum number λ, find a nearest child neuron of the nearest neuron at the next bottom level.

Step 4: Repeat step 3 until the found neuron is associated with the least number of images that is still more than the prespecified number λ. The last found neuron is a target one for next steps.

Step 5: Directly compare the distance between the query image and the target neuron by region-based features. Sort the images by distance with an ascending order and provide them to users.

Step 6: Users select some retrieved images by their semantic meaning. This information is fed back to front-end of the CBIR system. The old query is then modified to a new one according to the users' feedback. And the new query is supposed to retrieve more relevant images. This step is called relevance feedback (RF). RF is usually iterated for several times.

In the above retrieve procedure, the determination of the value of λ depends also on the number of input data. The more the number of input data, the larger the value of λ will be. Usually the value of λ can be taken as 1% to 5% of the number of input data for satisfactory performance.

The architecture of GHSOQM-based CBIR system is shown in Fig. 2. Note that the proposed CBIR system uses a hierarchical structure by GHSOQM to organize images and GHSOQM must be first trained by using all images. Retrieval processes in some CBIR systems, e.g., SIMPLIcity, directly compare query image with all images. It uses a flat structure and does not require any training. The extra work by GHSOQM is compensated by a faster retrieval time.

4.2. Image segmentation, feature extraction and region-based feature matrices

The image segmentation algorithm we used is the JSEG algorithm. JSEG first quantizes colors in an image and generates a class map. Based on the class map, JSEG finds a good segmentation with coarse or precise resolution by using a criterion for goodness of segmentation.

After image segmentation we can perform feature extraction for each region of an image. Thirteen features are extracted for each region, i.e., six for colors, six for textures and one for region percentages of images. In this paper, the *Lab* perceptually uniform color space is used, where *L* represents the luminance of the color, *a* represents the position between red and green, *b* represents the position between yellow and blue. We compute the average and standard deviation of the *L*, *a* and *b* components in the *Lab* color space for each region of an image. We denote the average of *L*, *a* and *b* as f_1, f_2, and f_3, the standard deviation of *L*, *a* and *b* as f_4, f_5, and f_6. For texture features, we first compute the following three variables for a 4×4 block in an image as used in Ref. [16]

$$a = \sqrt{\left(\sum_{i=1}^{2}\sum_{j=1}^{2} a_{ij}^2\right)\Big/ 4}, \quad b = \sqrt{\left(\sum_{i=1}^{2}\sum_{j=1}^{2} b_{ij}^2\right)\Big/ 4},$$
$$c = \sqrt{\left(\sum_{i=1}^{2}\sum_{j=1}^{2} c_{ij}^2\right)\Big/ 4}, \tag{5}$$

where $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, $\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$ and $\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$ are the coefficients of Haar wavelet transform for *LH*, *HL* and *HH* band, respectively. After wavelet transformation, we just assign the three variables to each pixel of the block. Then we compute the average and standard deviation of the three features *a*, *b* and *c* for each region. We denote the average of *a*, *b* and *c* as f_7, f_8 and f_9, the standard deviation of *a*, *b* and *c* as f_{10}, f_{11} and f_{12}. The last feature f_{13} is the region percentage of an image.

So an image x can be denoted by region-based features matrix

$$\begin{bmatrix} R_1^x \\ \dots \\ R_n^x \end{bmatrix},$$

where $R_i^x = [f_{i,1}^x, \dots, f_{i,12}^x]$ $(i = 1, \dots, n)$ is a row feature vector representing the ith region of the image x, n is the number of regions in the image. Different images may have different number of regions. For the sake of convenience, a neuron in the GHSOQM-based CBIR system is represented by a feature matrix with a fixed number of rows, which means a fixed number of regions.

4.3. Image distance

Since the representation of an image is a feature matrix, we defined a distance measure in order to compare the dissimilarity of two images. The IRM distance criterion compares two images with mutual directions. And the algorithm of weight assignment for regions in IRM is a little complicated. In this study, images are compared with direction from query image to other images. The weight assignment for each region of an image is just the region percentage of the image.

Suppose we have two images A and B. Image A has n regions and image B has m ones. The corresponding representing matrix are

$$\begin{bmatrix} R_1^A \\ \dots \\ R_n^A \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} R_1^B \\ \dots \\ R_m^B \end{bmatrix},$$

where R_i^A is the row feature vector of the ith region of image A and each component of all feature vectors are normalized to lie in [0, 1]. The region distance between R_m^A and R_n^B is defined as the following:

$$d_{mn} = \left(w_1 \sum_{i=1}^{3} (f_{m,i}^A - f_{n,i}^B)^2 + w_2 \sum_{i=4}^{6} (f_{m,i}^A - f_{n,i}^B)^2 + w_3 \sum_{i=7}^{9} (f_{m,i}^A - f_{n,i}^B)^2 + w_4 \sum_{i=10}^{12} (f_{m,i}^A - f_{n,i}^B)^2 \right)^{1/2}, \quad (6)$$

where w_1–w_4 are the weights to colors and textures. In this study, the weights are chosen such that $w_1 = w_3 = w_4 = 1$, and $w_2 = 0.5$. With this selection of weights for colors and textures, image retrieval results are satisfactory.

The distance from image A to B is described as the following steps.

Step 1: Compute the distance matrix D, where d_{mn} is the element in mth row and nth column of the matrix and denoted by Eq. (6).

Step 2: Find the minimum value in each row of the matrix D and denote D_i as the minimum of the ith row of D.

Step 3: Compute the weighted average for distance from image A to B:

$$Distance(A, B) = \sum_{i=1}^{n} f_{i,13}^A D_i. \quad (7)$$

4.4. GHSOQM and relevance feedback in the proposed CBIR system

The GHSOQM algorithm is used in the proposed CBIR system. Images and weights of neurons are represented by feature matrices. A large number of computations in GHSOQM is to find the nearest neurons to retrieve images. As mentioned before, a neuron i represents an image at time t by

$$w_i = \begin{bmatrix} R_1^i(t) \\ \dots \\ R_r^i(t) \end{bmatrix},$$

where r is the fixed number of regions. The distance from an image A to neuron i (with weight matrix w_i) is the same function as (7):

$$Distance(A, w_i) = \sum_{i=1}^{n} f_{i,13}^A D_i, \quad (8)$$

where D_i is the minimum value in the ith row of the distance matrix D between image A and neuron i.

The weight updating for neurons must be modified in the proposed CBIR system because of the matrix representation of images. The weight updating now is the following steps:

Step 1: Find the nearest regions of an updating neuron k from a query image

$$x = \begin{bmatrix} R_1^x \\ \dots \\ R_n^x \end{bmatrix}$$

at time t. The found regions of the neuron are arranged with order in a matrix

$$\begin{bmatrix} R_1^k(t)' \\ .. \\ R_n^k(t)' \end{bmatrix}$$

corresponding to the regions of x. Note that the found regions may repeat such that $R_i^k(t)'$ and $R_j^k(t)'$ are the same as a region of the neuron k.

Step 2: Like (4), update the neuron by

$$R_i^k(t+1)' = R_i^k(t)' + \varepsilon(t) h_{kc}(R_i^x - R_i^k(t)'), \quad i = 1, \dots, n, \quad (9)$$

where h_{kc} is the neighborhood function, R_i^x is the feature vector of the ith region of the query image x, $R_i^k(t)'$ is the feature vector of the nearest region in neuron k at time t from the ith region of the query image x.

For example, a query image x has three regions

$$\begin{bmatrix} R_1^x \\ R_2^x \\ R_3^x \end{bmatrix}$$

and an updating neuron k have four

$$\begin{bmatrix} R_1^k \\ R_2^k \\ R_3^k \\ R_4^k \end{bmatrix}.$$

The corresponding nearest regions of the neuron from the query image are

$$\begin{bmatrix} R_2^k \\ R_3^k \\ R_3^k \end{bmatrix}$$

with order. The weight updating is

$$\begin{aligned} R_2^k(t+1) &= R_2^k(t) + \varepsilon(t)h_{kc}(R_1^x - R_2^k(t)), \\ R_3^k(t+1) &= R_3^k(t) + \varepsilon(t)h_{kc}(R_2^x - R_3^k(t)), \\ R_3^k(t+1) &= R_3^k(t) + \varepsilon(t)h_{kc}(R_3^x - R_3^k(t)). \end{aligned} \tag{10}$$

Note that region 3 of the neuron k is updated twice because it is the nearest region from regions 2 and 3 of the query image.

RF of region-based feature matrices is like that of feature vectors in Eq. (1). Assume the query image has a feature matrix

$$x = \begin{bmatrix} R_1^x \\ \cdots \\ R_n^x \end{bmatrix}.$$

The retrieved images are classified as relevant images $Y = \{Y_1, \ldots, Y_{|Y|}\}$ and irrelevant images $Z = \{Z_1, \ldots, Z_{|Z|}\}$. For image Y_i, find the nearest regions from the regions of the image x and denote them as a matrix

$$Y_i' = \begin{bmatrix} Y_{i1}' \\ \cdots \\ Y_{in}' \end{bmatrix},$$

where Y_{ij}' is the nearest region of image Y_i from the jth region of the query image x. Similarly, the nearest regions from the image x are denoted as a matrix

$$Z_i' = \begin{bmatrix} Z_{i1}' \\ \cdots \\ Z_{in}' \end{bmatrix}$$

for each Z_i. Then the new query matrix is modified by

$$R_i^x(t+1) = \alpha R_i^x(t) + \frac{\beta}{|Y|}\sum_{k=1}^{|Y|} Y_{ki}' - \frac{\gamma}{|Z|}\sum_{k=1}^{|Z|} Z_{ki}', \quad i = 1, \ldots, n, \tag{11}$$

where α, β and γ are parameters controlling the relative weighting of current query image, relevant images and irrelevant images, respectively.

4.5. Incremental learning of GHSOQM

For the new added images, the already learned neural network must have the ability of learning new image without reusing old images. The incremental learning of GHSOQM implements this objective. The concrete steps of incremental learning are described in the following:

Step 1: Assign the new images to the neurons level by level as the image assignment in GHSOQM.

Step 2: Check the leaf neurons if they can generate child neurons or not. If the number of data associated with a leaf neuron is more than the pre-specified number τ, the neuron has to generate four child neurons. Train the weights of the four new neurons by $TRAIN\_SOM(X, n, W)$ and generate child SOMs by recursively calling $GENERATE\_SOM(X, n, W)$ if possible as described above, where X are the data belonging to the leaf neuron at level n after new data are added, W is the weights of the new four neurons at level $n+1$ to be trained.

In a word, the basic principle of incremental learning of GHSOQM is to assign the new images into the existed network and grow new neurons hierarchically for the old leaf neurons, with which the number of images associated is more than the pre-specified number τ.

5. Experimental results

In this section, 1000 images were firstly used to test the effectiveness of the proposed CBIR system. Then 500 new images were added to the system later without retraining all images. Finally the robustness of the system was tested for some alterations of sample images. Our testing system was implemented on a Pentium III 733 MHz PC using MATLAB software. The number of fixed regions for representing a feature image of one neuron is set to 10. After image retrieval, users select relevant images by checking in the boxes on the upper left corners of the relevant images. The next new query by RF leaves out already checked images. And all checked images are always listed on the top rank.

5.1. Performance of the GHSOQM-based CBIR system on a static images data set

In this subsection, 1000 images [16] were evaluated to test the proposed system. The images have ten classes, each

Table 1
Ten classes of 1000 experimental images

Classes	Semantic name
1	African people and village
2	Beach
3	Building
4	Buses
5	Dinosaurs
6	Elephants
7	Flowers
8	Horses
9	Mountains and glaciers
10	Food

of which contains 100 pictures. The sizes of the images are 384×256, or 256×384. The ten classes are listed in Table 1. The parameter τ for GHSOQM training was set to 20, which means that a neuron must generate its child neurons if the number of images belonging to it is more than 20. Another parameter λ for the retrieval process was set to 20, which means that the least number of a target neuron for image retrieval is more than 20. Furthermore, the implemented GHSOQM-based CBIR system only shows the first 20 images to users. After completion of training GHSOQM, the CBIR system was ready for testing.

First, the proposed CBIR system requires training for GHSOQM. By using MATLAB software, the average training time by GHSOQM is 1890 seconds for the 1000 images while the SIMPLIcity system does not require any training. But GHSOQM-based query makes the querying time faster than direct query without training. By using MATLAB software, the average search time of GHSOQM-based query for one image is 3.08 s and direct query is 7.25 s. The proposed query is much faster than direct query because the hierarchical structure of images and the additional training before querying.

Next we compare the recall-precision graph of the proposed query and direct query. Precision P is defined as the following:

$$P(k) = n_k/k, \tag{12}$$

where k is the number of retrieved images and n_k is the number of relevant images in the retrieved images. Recall R is defined as

$$R(k) = n_k/N, \tag{13}$$

where N is the number of all relevant images in the data set. An optimal recall-precision graph would have a straight line, i.e., precision always at 1. Typically, when recall increases, precision decreases accordingly. Since the proposed system only shows the first 20 images, the maximum value of recall is 0.2 in this study. We used the aforesaid 10 sample images from all classes and tested the performance of GHSOQM-based query and direct query. The recall-precision graphs are plotted in Fig. 3, where GHSOQM-based query and direct

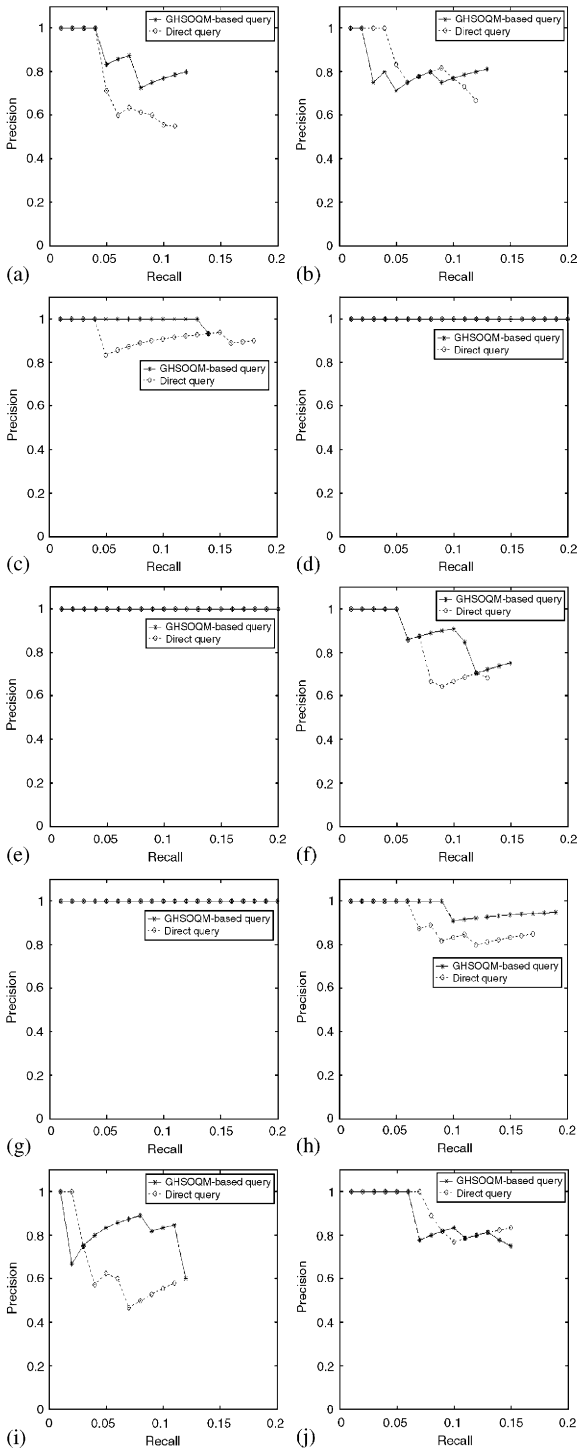


Fig. 3. Recall-precision graphs for GHSOQM-based query and direct query on the 10 images (a) 097.jpg (African people and village), (b) 173.jpg (beach), (c) 219.jpg (building), (d) 325.jpg (buses), (e) 411.jpg (dinosaurs), (f) 586.jpg (elephants), (g) 672.jpg (flowers), (h) 788.jpg (horses), (i) 861.jpg (mountains and glaciers), (j) 906.jpg (food).

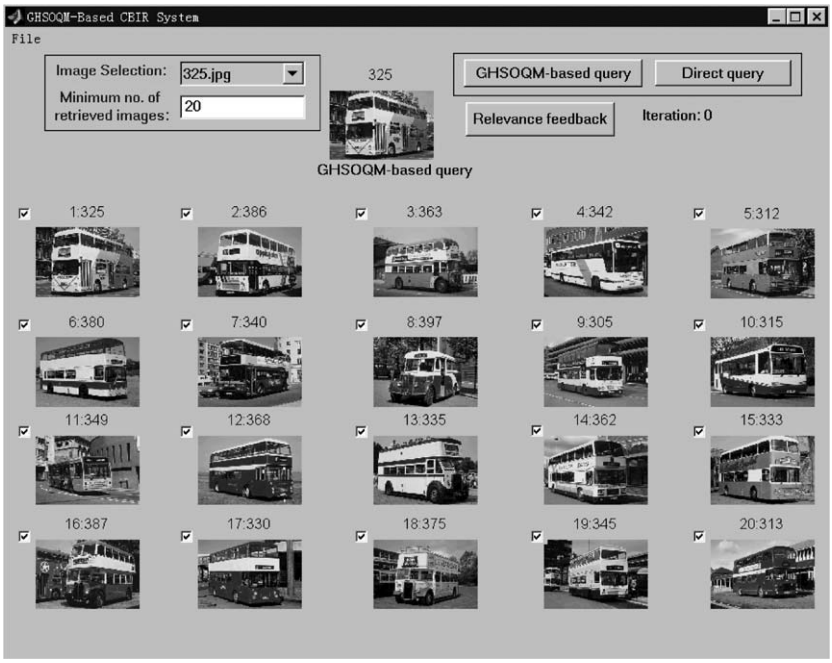


Fig. 4. Retrieval results of image 325.jpg for GHSOQM-based query, corresponding to Fig. 3.

query have similar query performance. The corresponding retrieval result of Fig. 3d is shown in Fig. 4. In Fig. 3, the recall-precision graphs of some images are optimal at the recall interval [0 0.2], i.e., buses, dinosaurs and flowers. This is because the objects in these classes have simpler color distributions. The performance for the images from other classes is degraded because the objects in the images have more complex color distributions. We also tested some other images in each class and obtained similar results like Fig. 3.

Combined with RF, the proposed CBIR system obtained more relevant images than that with one-pass query. The parameters of relevance feedback in this study are such that $\alpha = 0.95$, $\beta = 0.05$, $\gamma = 0.05$ in (11). For RF, the improved performance is evaluated by precision defined as the percentage of relevant image in the 20 retrieved images, i.e., $P(20)$ in (12). We tested the proposed CBIR system on the previous sample images without optimal recall-precision graphs in Fig. 3. The results with RF for GHSOQM-based query and direct query are shown in Fig. 5a–g. Almost all queries on the sample images have been improved by RF. The average precision for GHSOQM-based query is increased by about 0.18, while that for direct query is increased by about 0.15. The example image 788.jpg in horse class achieved 100% precision after 5 iterations of RF for both types of queries as shown in Fig. 5e.

GHSOQM organizes the images in a hierarchical level. At top level, a large fraction of images of different classes are mixed up into one neuron. A coarse resolution map is thus formed. As with the level increased, the map becomes more and more precise with a dominant class. The histograms of images belonging to neurons at levels 1 and 2 are plotted in Fig. 6, where no single classes are dominant for any neuron at level 1. The histograms of images associated with the child neurons at level 2 are depicted at the four corners in Fig. 6. Some of the histograms have single dominant class. For examples, the first neuron at level 2 at the upper right corner, whose parent is the second neuron at level 1, has a dominant class 5, which is mixed up with other classes in the second neuron at level 1. As another example, the fourth neuron at level 2 at bottom left corner, whose parent is the third neuron at level 1, has a dominant class 4. It is the single dominant class that makes the retrieved result with high precision.

5.2. Dynamic performance of incremental learning of the GHSOQM-based CBIR system with new images

In this case, 500 new images were added into the old images. The sizes of the new images are 213×160 or 160×213. The added images can be roughly classified into forest, mountain, flowers, sunset, blue sky, trees, grasses, beach, snowed land, etc. Some classes are included in the old images and some are quite new. We used the proposed incremental learning of GHSOQM on the 500 new images and GHSOQM on all 1500 images. By using MATLAB software, the average training time of the first type of training is 1046 s while that of the second type is 2664 s. The time saved is about 1618 s. The incremental property in the

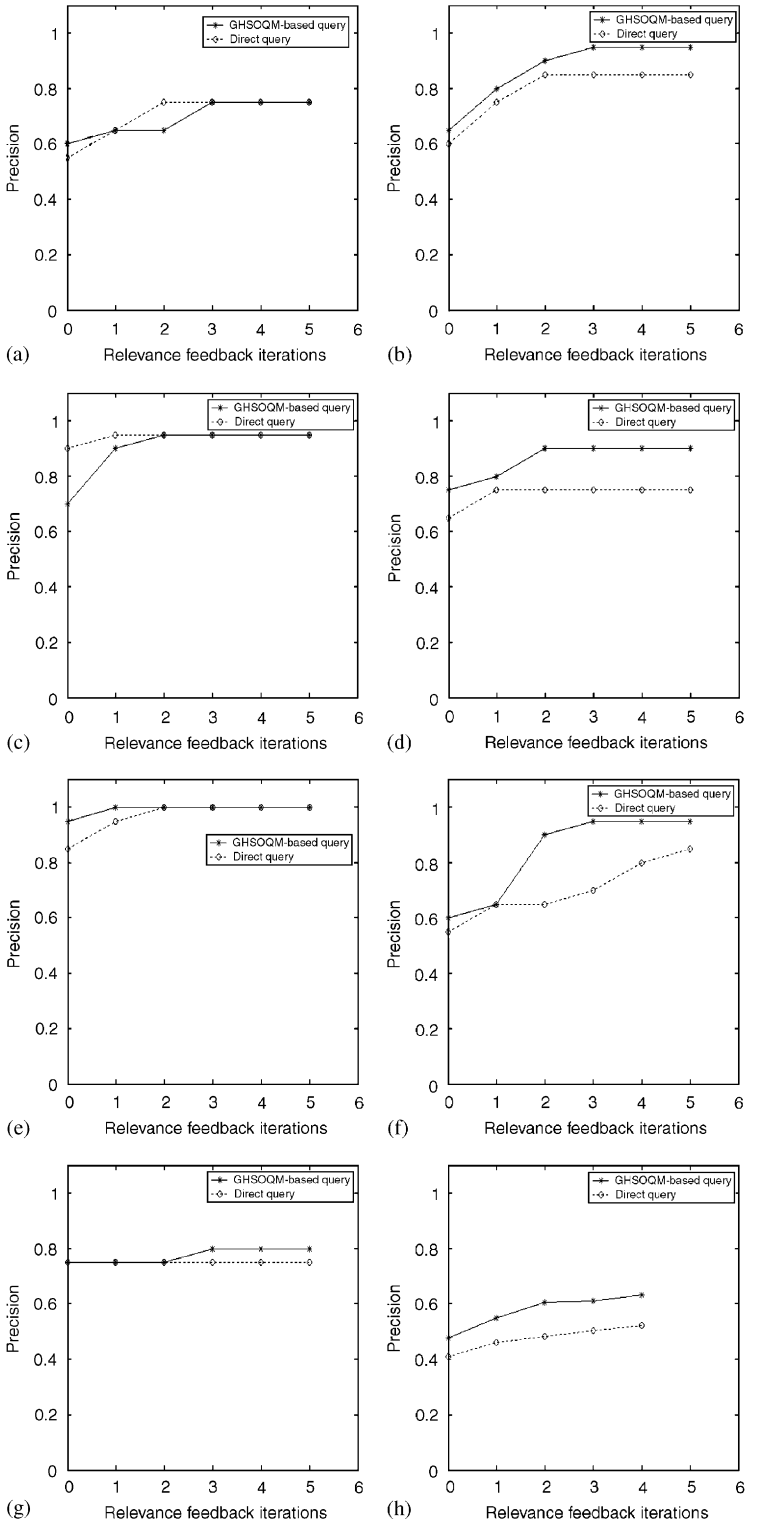


Fig. 5. Precision with iterations of relevance feedback between GHSOQM-based query and direct query for (a) 097.jpg, (b) 173.jpg, (c) 219.jpg, (d) 586.jpg, (e) 788.jpg, (f) 861.jpg, (g) 906.jpg, (h) 500 new images (average).

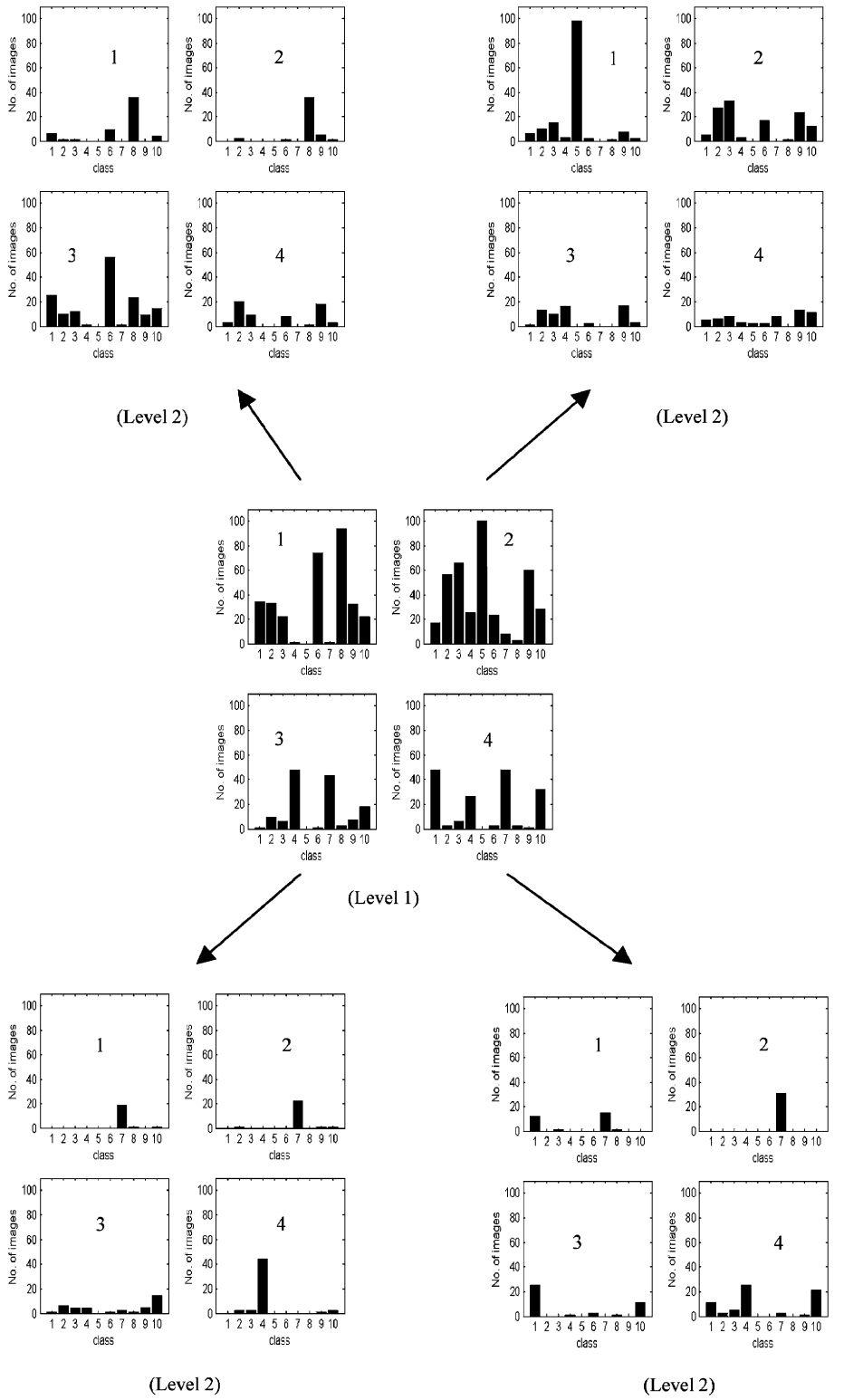


Fig. 6. Statistical results of images associated with neurons at levels 1 and 2.

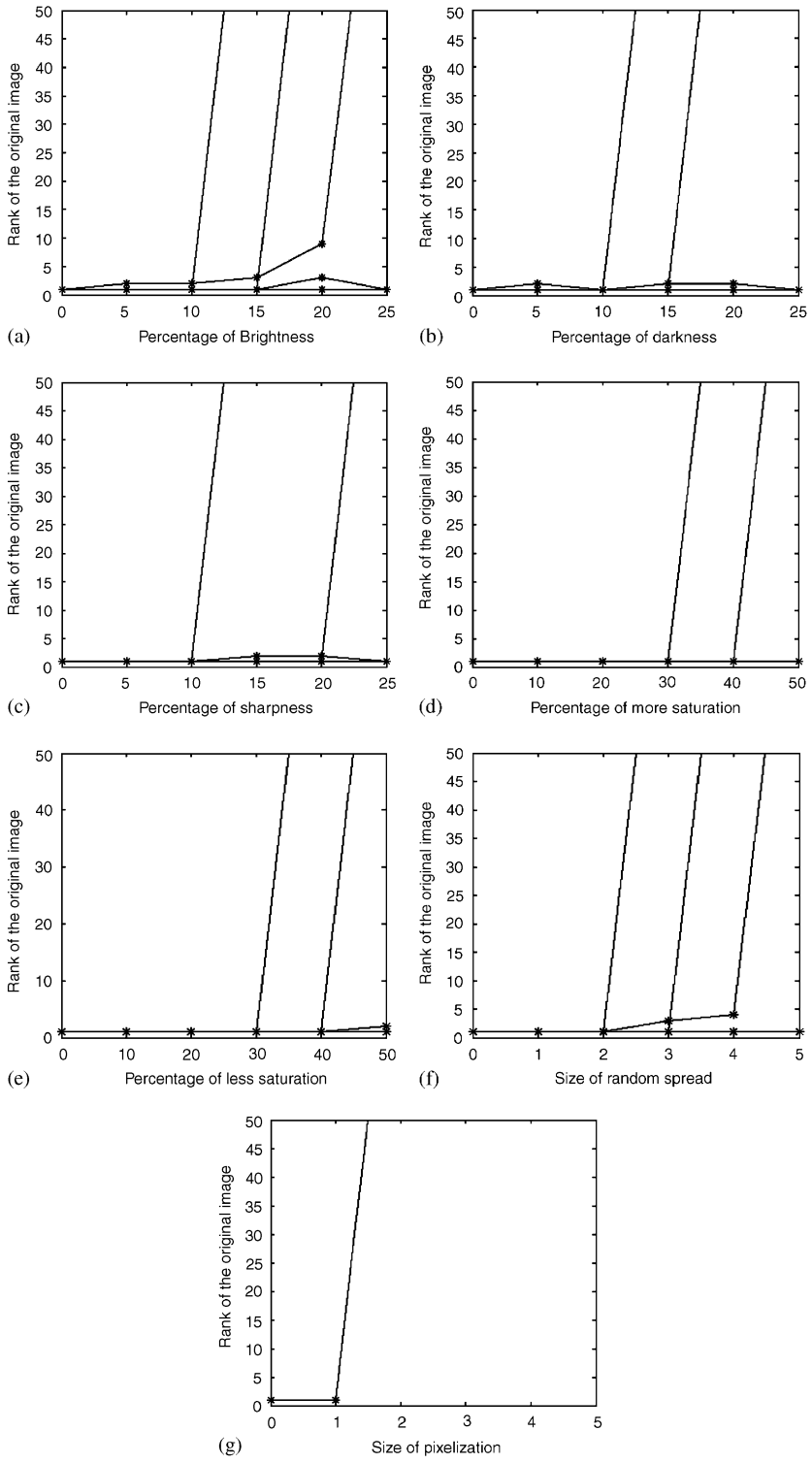


Fig. 7. Rank of the original image in the retrieved images for some image alterations on six sample images by GHSOQM-based query: (a) brightness, (b) darkness, (c) sharpness, (d) more saturation, (e) less saturation, (f) random spread, (g) pixelization.

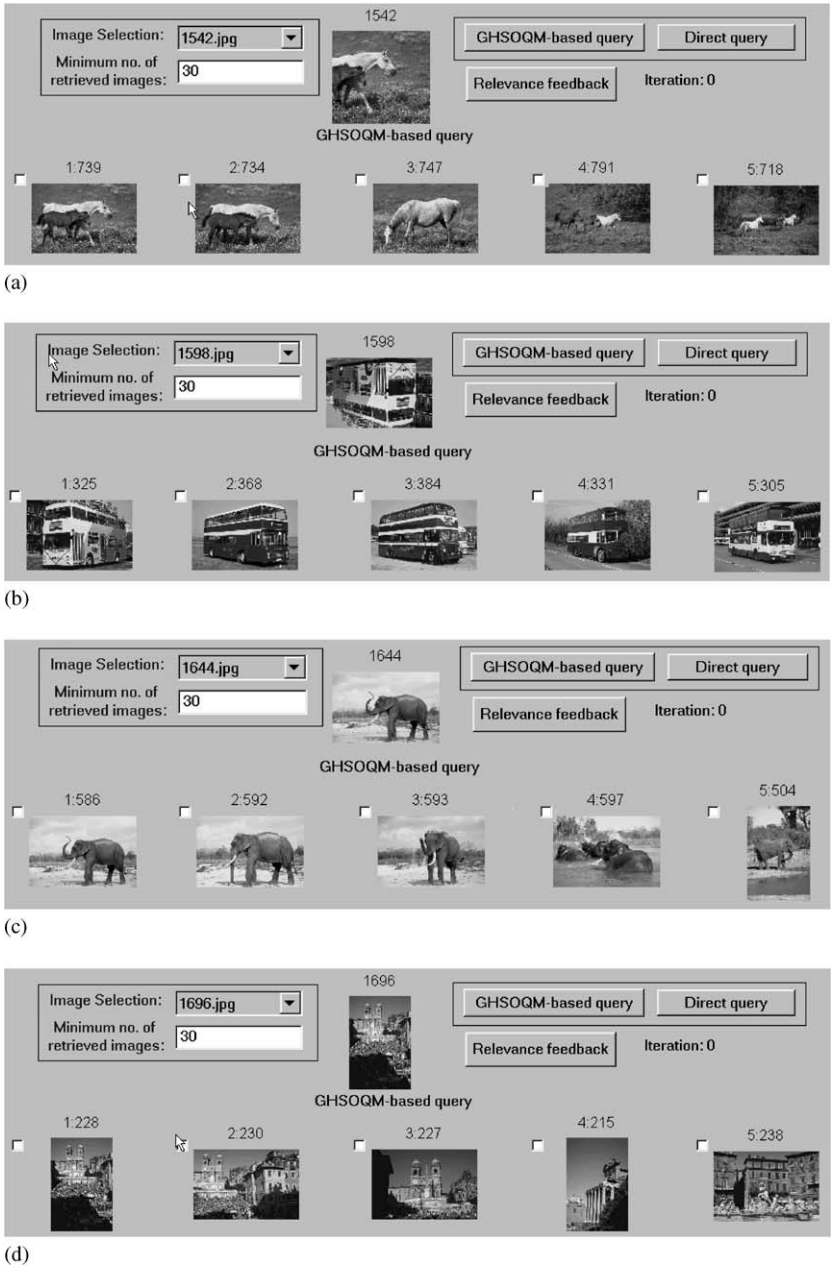


Fig. 8. Retrieval results of altered images: (a) cropping (1542.jpg) from original image (739.jpg), (b) flipping (1598.jpg) from original image (325.jpg), (c) horizontal shifting 2% (1644.jpg) from original image (586.jpg), (d) vertical shifting 2% (1696.jpg) from original image (228.jpg).

proposed CBIR system is very important because retraining with all data is not practicable and is wasteful.

We randomly selected 9 images from the new images and tested the new incrementally learned CBIR system. The average precision with iterations of relevance feedback is shown in Fig. 5h. The average precision for GHSOQM-based query is increased by about 0.15, while that for direct query is increased by about 0.05. The GHSOQM-based query has similar precision results with direct query. But it needs less querying time.

5.3. Robustness of the GHSOQM-based CBIR system to image alterations

The robustness of the GHSOQM-based CBIR system is tested by altering some sample images, i.e., intensity

variation, sharpness, blur, saturation, distortion, cropping, shifting, rotating, etc. Six sample images from the first 1000 images were selected for such alterations. The robustness of the proposed query is reflected by the rank of the original image in the retrieved images. The higher the rank of the retrieved original image, the more robust the altered images will be. First we tested the seven types of alterations on the six sample images from light to heavy degree. The ranks of the retrieved original image with the variation of degree are shown in Fig. 7. Generally, the proposed system is robust to 10% brightening, 10% darkening, 10% sharpening, 30% more saturation, 30% less saturation, random spread by 2 pixels, and pixelization by 1 pixel. Fig. 8 shows such robustness to some sample images.

6. Conclusion

A framework of GHSOQM-based CBIR system was developed. The system has three advantages. First, the system needs a neural network, called GHSOQM, to learn images and thus organized images into a hierarchical structure. The hierarchical maps are from coarse top level to precise bottom level. Second, the query time is largely reduced due to the tree structure of GHSOQM. This is very useful for large image data sets today. Finally and most importantly, the learning of GHSOQM can be incremental such that only new images can be used for new training of GHSOQM if new images are not included in the system.

In order to represent region-based images, we proposed region-based feature matrices. We also proposed corresponding GHSOQM updating rule for the feature matrices of neurons. The mechanism of RF used in the proposed system improves the performance of image retrieval. Experimental results demonstrate the effectiveness and robustness of our proposed system.

Acknowledgements

The authors would like to thank the editor and the reviewers' useful comments. The work described in this paper was fully supported by a grant from the Research Grant Council of Hong Kong SAR Region [Project No. 7001599-570].

References

[1] Y. Rui, T.S. Huang, S.-F. Chang, Image retrieval: current techniques, promising directions, and open issues, J. Visual Commun. Image Representation 10 (1) (1999) 39–62.

[2] M. Flicker, H. Sawhney, W. Niblack, J. Ashley, et al., Query by image and video content: the QBIC system, IEEE Computer 28 (9) (1995) 23–32.

[3] J.R. Smith, C.S. Li, Image classification and querying using composite region templates, Int. J. Comput. Video Database 75 (1–2) (1999) 165–174.

[4] A. Gupta, R. Jain, Visual information retrieval, Comm. ACM 40 (5) (1997) 70–79.

[5] S. Mukherjea, K. Hirata, Y. Hara, AMORE: a world wide web image retrieval engine, World Wide Web 2 (3) (1999) 115–132.

[6] A. Natsev, R. Rastogi, K. Shim, WALRUS: a similarity retrieval algorithm for image database, SIGMOD Record 28 (2) (1999) 395–406.

[7] A. Pentland, R.W. Picard, S. Sclaroff, Photobook: tools for content-based manipulation of image databases, Proc. SPIE 2185 (1994) 34–37.

[8] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, J. Malik, Blobworld: a system for region-based image indexing and retrieval, in: Third International Conference on Visual Information Systems, Amsterdam, The Netherlands, June 1999, pp. 509–516.

[9] J.R. Smith, S.-F. Chang, VisualSEEK: a fully automated content-based image query system, in: Proc. ACM Multimedia, Boston, MA, USA, November 1996, pp. 87–98.

[10] S. Stevens, M. Christel, H. Wactlar, Informedia: improving access to digital video, Interactions 1 (4) (1994) 67–71.

[11] W.Y. Ma, B. Manjunath, NeTra: a toolbox for navigating large image databases, ACM Multimedia Syst. 7 (3) (1999) 184–198.

[12] R. Jain, S.N.J. Murthy, P.L.-J. Chen, S. Chatterjee, Similarity measures for image databases, Proc. SPIE 2420 (1995) 58–65.

[13] E.G.M. Petrakis, A. Faloutsos, Similarity searching in medical image databases, IEEE Trans. Knowledge and Data Eng. 9 (3) (1997) 435–447.

[14] Y. Rubner, L.J. Guibas, C. Tomasi, The earth mover's distance, multi-dimensional scaling, and color-based image retrieval, in: Proc. DARPA Image Understanding Workshop New Orleans, LA, USA, May 1997, pp. 661–668.

[15] J.Z. Wang, G. Wiederhold, O. Firschein, X.W. Sha, Content-based image indexing and searching using Daubechies' wavelets, Int. J. Digital Libraries 1 (4) (1998) 311–328.

[16] J.Z. Wang, J. Li, G. Wiederhold, SIMPLIcity: semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Mach. Intelligence 23 (9) (2001) 947–963.

[17] P. Koikkalainen, Fast deterministic self-organizing maps, Proceedings of the International Conference on Artificial Neural Networks, vol. 2, Paris, France, October 1995, pp. 63–68.

[18] T. Kohonen, Self-Organizing Maps, Springer, Berlin, Germany, 1997.

[19] J. Laaksonen, M. Koskela, S. Laakso, E. Oja, PicSOM-content-based image retrieval with self-organizing maps, Pattern Recogn. Lett. 21(13–14) 1199–1207.

[20] J. Laaksonen, M. Koskela, E. Oja, PicSOM- self-organizing image retrieval with mpeg-7 content descriptors, IEEE Trans. Neural Networks 13 (4) (2002) 841–853.

[21] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, USA, 1983.

[22] B. Fritzke, Growing cell structure: a self-organizing network for supervised and unsupervised learning, Neural Networks 7 (9) (1994) 1441–1460.

[23] B. Fritzke, A growing neural gas network learns topologies, in: G. Tesauro, D.S. Touretzky, T.K. Leen (Eds.), Advances in Neural Information Processing Systems 7, MIT Press, Cambridge, MA, 1995, pp. 625–632.

[24] S. Marsland, J. Shapiro, U. Nehmzow, A self-organizing network that grows when required, Neural Networks 15 (8–9) (2002) 1041–1058.
[25] J. Si, S. Lin, M.-A. Vuong, Dynamic topology representing networks, Neural Networks 13 (6) (2000) 617–627.
[26] H.-U. Bauer, T. Villmann, Growing a hypercubical output space in a self-organizing feature map, IEEE Trans. Neural Networks 8 (2) (1997) 218–226.
[27] D. Alahakoon, S.K. Halgamuge, B. Srinivasan, Dynamic self-organizing maps with controlled growth for knowledge discovery, IEEE Trans. Neural Networks 11 (3) (2000) 601–614.
[28] J. Blackmore, R. Miikkulainen, Visualizing high-dimensional structure with the incremental grid growing neural network, in: Proceedings of the 12th International Conference on Machine learning, Tahoe City, CA, USA, July 1995, pp. 55–63.
[29] H. Samet, The quadtree and related hierarchical data structures, ACM Comput. Surveys 16 (2) (1984) 187–260.
[30] S.-B. Cho, Self-organizing map with dynamical node splitting: application to handwritten digit recognition, Neural Computation 9 (6) (1997) 1345–1355.
[31] T.W.S. Chow, S. Wu, Cell-splitting grid: a self-creating and self-organizing neural network, Neurocomputing 57 (2004) 373–387.
[32] P. Muneesawang, L. Guan, Automatic machine interactions for content-based image retrieval using a self-organizing tree map architecture, IEEE Trans. Neural Networks 13 (4) (2002) 821–834.
[33] J. Herrero, A. Valencia, J. Dopazo, A hierarchical unsupervised growing neural network for clustering gene expression patterns, Bioinformatics 17 (2) (2001) 126–136.
[34] A. Rauber, D. Merkl, M. Dittenbach, The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data, IEEE Trans. Neural Networks 13 (6) (2002) 1331–1341.
[35] Y. Deng, B. S. Manjunath, H. Shin, Color image segmentation, Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99), vol. 2, Fort Collins, CO, June 1999, pp. 446–451.

About the Author—SITAO WU received Ph.D degree in the Department of Electronic Engineering of City University of Hong Kong, Hong Kong in 2004. He received B.E. and M.E. degrees in the Department of Electrical Engineering of Southwest Jiaotong University, China in 1996 and 1999, respectively. His research interest areas are neural networks, pattern recognition, and their applications.

About the Author—M.K.M. RAHMAN received his B.Eng. degree in the Department of Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology in 2001. He is currently working towards the Master of Philosophy degree at City University of Hong Kong, Hong Kong. His research interests are structural data processing, neural network, image retrieval and classification.

About the Author—TOMMY W.S. CHOW received the B.Sc. (1st Hons) degree. He undertook his Trainee with Reyrolle Technology, U.K. He received the Ph.D. degree from the University of Sunderland, England, U.K., working on a collaborative project between The International Research and Development, Newcastle Upon Tyne, U.K., and the Ministry of Defense (Navy) U.K.

Following receipt of the Ph.D. degree, he joined the City University of Hong Kong, where currently he is a Professor in the Department of Electronic Engineering. He has been a consultant to the Mass Transit Railway, Kowloon-Canton Railway Corporation, Hong Kong. He has conducted other collaborative projects with the Kong Electric Co. Ltd., and Royal Observatory Hong Kong, and the MTR Hong Kong on the application of neural networks for machine fault detection and forecasting. One of these works also led to the Best paper award in the 2002 IEEE Industrial Electronics Society Annual meeting in Seville, Spain. His main research has been in the area of neural network, learning theory, system identification, and machine fault diagnostics. He is an author and co-author of numerous published works, including book chapters, and over 90 Journal articles related to his research.