Visual Thesaurus for Color Image Retrieval using Self-Organizing Maps

Christopher C. Yang and Milo K. Yip
Department of System Engineering and Engineering Management
The Chinese University of Hong Kong, Hong Kong

ABSTRACT
The technique of searching in content-based image retrieval has been actively studied in recent years. However, this technique cannot give user an overview of the database. In this paper, we propose a browsing technique using Kohonen’s Self-Organizing Map to retrieve general color image database effectively. Both chromatic and textural feature of images are analyzed to represent the content of images.

Keywords: Visual database, Content-based Retrieval, Image Browsing, Self-Organizing Map, Color Histogram, Gabor Filter

1. INTRODUCTION
Recently, many approaches for content-based image retrieval (CBIR) have been developed; for example, IBM’s Query by Image Content (QBIC), and UC Santa Barbara’s Alexandria Digital Library (ADL) project. Most of them use the technique of query-by-example, a searching approach to find images that are similar to the given example image. Similarity between images is quantified in terms of some global or local features, such as colour, texture, shape, pattern, or specific spatial area of the above. The feature selection and matching techniques are vital to searching thus much research worked on these techniques. However, there are some serious limitations on using searching solely. For example, query-by-example techniques often generate results of a relatively small number of images, which may not be interested by the user; as a result, the user may not be able to make further querying. In contrast, browsing environment offer an alternate approach to the problem, but it has received relatively little attention.

According to Craverl et al. [1], browsing is a technique, or a process, that the users view information rapidly and decide whether the content is relevant to their needs. The user can get an overview or summary of content quickly and then focus on particular sections of interest. Craver et al. modeled a general process of image browsing. The first step is to extract the relationship among the images. The second step is to find representative instances of images during browsing. The final step is to visualize the images with intuitive presentation of the relationship among images.

Recent research in images browsing uses different data structure to organize images in a meaningful way to present the relationship among images. Craver et al. described a technique based on multiple space-filling curves. Chen et al. [2] initiated a new approach called active browsing. The research made use of their previous work on similarity pyramids. Those techniques provide an attractive feature that the process of insertion of new images is computationally fast; complete re-indexing is not necessary.

We investigate the unsupervised artificial network, Kohonen’s self-organizing map (SOM) [3, 4], for image browsing. SOM reduces high-dimensional input signal space into low-dimensional space while the map preserves the relationship among the input signals.

Zhang et al. [5] addressed the importance of developing effective indexing scheme for image database searching. They studied an indexing scheme that applies SOM approach. Three texture features – MRSAR, coarseness and gray histogram – were used for indexing monochrome textural images and color histogram was used for indexing general color images. They developed a set of hierarchical self-organizing maps (HSOMs) to construct an index tree, which provides a space for searching. The evaluation of performance is calculated by retrieval rate.

Han et al. [6] uses SOM for image retrieval. Their system used SOM for generating maps for human browsing but not for machine indexing. However, their system is restricted to handle images of objects, from which shape features such as roundness, rectangularity, ellipticity, eccentricity and bending energy are extracted, where only the luminance information is considered. Besides, they used a set of randomly selected images to label the regions in the map, which is not necessary representing the most significant features of the node.

In this paper, we present an approach to provide a image-browsing interface for human to retrieve general colour image database using SOM. We have tested advanced chromatic and texture features to generate maps and constructed three methods to label the resultant maps.

2. ALGORITHM
Our proposed algorithm can be divided into two main sections: image representation and self-organizing map generation. The former section transforms the image database into representation, which can be used for evaluating similarity among the images. The latter section generates SOM using the above information.

Image Representation
There are a number of colour coordinate systems for images. Traditionally colour images are represented in RGB format when processed in computer systems. However, this colour coordinate system does not match with human perception of colour. Colour perception research generally believe that human recognize colour by hue and saturation. Since this research is aimed at providing an interface for human, we have adopted LHS color coordinate system that corresponds to human color perception.

The process of the proposed algorithm for image representation is shown as follows.
Feature Analysis

where number of pixels in the image. The definition analysis. The algorithm is to build a two-dimensional histogram as mentioned before, we use H and S values for chromatic occlusion, the histogram only changes slightly.

The hue component varies from 0 to 360 degrees and the saturation component varies from 0 to 1. In order to build the histogram, the hue and saturation values are quantized into several levels. We have chosen 10 levels for each component, so that a 10 by 20 two-dimensional histogram \( h[10, 10] \) for each image is built in our system.

A Pixel with \( 0^\circ \leq h < 36^\circ \) and \( 0 < s < 1 \) is sorted in the first bin \( h[1, 1] \) and so on.

Textural Analysis: There is a long history for texture analysis research. According to Manjunath et al. [8], texture analysis algorithms use various techniques, including random field model and multiresolution filtering techniques such as wavelet transform. In the research of Manjunath et al., the experiment results of a number of texture features are compared. The texture features includes the conventional pyramid-structured wavelet transform (PWT) features, tree-structured wavelet transform (TWT) features, and the multiresolution simultaneous autoregressive model (MR-SAR) features and the Gabor wavelet features. The experimental results show that Gabor features give the best performance.

In another recent research carried out by Chen et al. [9], they compared four filtering techniques including Fourier transform, spatial filter, Gabor filter and wavelet transform for texture discrimination. The experimental results also show that Gabor filter generally gives the best performance. However, Chen et al. stated that the execution time of Gabor filter is the longest out of the four features.

Since texture analysis are generated once only in this kind of image retrieval application, the disadvantage of longer execution time of generating Gabor features is not critical. Therefore, we adopt Gabor filter for textural analysis in this research due to its best performance. In the following, we will review the Manjunath et al.’s method of using Gabor filter for texture feature extraction.

A two-dimensional Gabor function can be written as:

\[
g(x, y) = \left( \frac{1}{2\pi \sigma_x \sigma_y} \right) \exp \left[ -\frac{1}{2} \left( \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\pi Wx \right]
\]  

(2)

Then a self-similar filter dictionary can be obtained as a mother Gabor Wavelet \( G(x, y) \) by appropriate dilations and rotations of Eq. (2) as:

\[
G_{mn} = a^{\frac{m-n}{2}} G(\theta_m, \theta_n)
\]

where \( h = \text{height of image}, w = \text{width of image}, \)
\( hside = (h - 1)/2, wside = (w - 1)/2 \)
\( \theta_m = (x - hside) \cos(n\pi / K) + (y - wside) \sin(n\pi / K) \)
\( \theta_n = -(x - hside) \sin(n\pi / K) + (y - wside) \cos(n\pi / K) \)
\( a > 1, m, n \text{ are integers} \)

Given an image with luminance \( f(x, y) \), a Gabor decomposition can be obtained by multiplying the luminance by the magnitude of the Gabor Wavelet:
The mean and standard deviation of the magnitude of the transform coefficient are used to represent the texture feature for classification and retrieval purpose:

\[
\mathbf{W}_{mn}(x, y) = f(x, y) \sqrt{G_{xx} \sigma^2 + G_{yy} \mu^2}
\]  

(4)

The input patterns are the feature vectors of images in the database. Each node in input layer (component of feature vector) is fully connected to the Kohonen Layer.

**Training:** The notations and training algorithm of SOM used in this research is as follows.

1. **Set of all images in database**
   
   I
   
   2. **Feature vector of image** \( i \in I \)
   
   \( x_i \)
   
   3. **Set of all nodes in Kohonen Layer**
   
   M
   
   4. **Weight vector of node** \( j \in M \) at time \( t \)
   
   \( w_j(t) \)
   
   5. **Set of images that are mapped to node** \( j \in M \)
   
   \( R_j \)
   
   6. **Label image of node** \( j \in M \)
   
   \( v_j \)

1. Randomize \( w_j(0), v_j \in M \)
2. For each iteration,
   
   2.1 Shuffle the presentation order of images
   
   2.2 For each image in the ordering list,
   
      2.2.1 Find a winning node
   
      2.2.2 Update the winning node and its neighbors
   
   2.3 Continue training with decreased learning rate and reduced area of neighborhood function
3. For each \( i \in I \), put \( i \) into \( R_j \)

Firstly, the weights of all nodes are assigned with a vector of random values. If there are identical weights of node, there may be a case that there are two winning nodes. To prevent this situation, the random process is used.

In each iteration, all images will be presented to the system. However, if the sequence of presentation of nodes is fixed, the map will be influence heavily by the first few images. Therefore, the order of presentation is shuffled in the beginning of each iteration to minimize this effect. When an image \( i \) is presented to the system, the feature vector \( x_i \) is compared with all weight vector \( w_j(t) \) of all nodes in Kohonen Layer. The most similar node, called the winning node, can be found. Similarity is defined by Euclidean distance. Denoting the winning node as \( c_i(t) \), finding a winning node is formulated as:

\[
c_i(t) = \arg \min_{c \in M} \| x_i - w_c(t) \|
\]

(8)

Then the winning node \( c_i(t) \) and its neighbors are updated. The objective of updating is to make the weights of the winning node and its neighbors become more similar to that input vector. The formula and the illustration of the vectors are shown as follow.

\[
w_i(t+1) = w_i(t) + \eta h_c(t) (x_i - w_i(t))
\]

(9)

where \( 0 < \eta < 1 \) is the learning factor and \( h_c(t) \) is the neighborhood function.

For convergence it is necessary \( \eta h_c(t) \rightarrow 0 \) when \( t \rightarrow \infty \). In this research, the learning factor \( \eta \) is a linearly decreasing function, of which the initial and final values are specified explicitly.

In this research, we use a constant weighting for the neighborhood function. The area of it diminished with respect to time.
where $N_c(t)$ refers to a neighborhood set of array points around node $c$ at time $t$. It has been depicted as:

$$h_c(t) = \begin{cases} 1 & \text{if } i \in N_c(t) \\ 0 & \text{if } i \notin N_c(t) \end{cases}$$ (10)

We defined the length of the side of square shape neighborhood set as:

$$\text{length}(l) = \max\{\max(SOM_{\text{width}}, SOM_{\text{height}}) \cdot (1 - t)^3\}$$ (11)

At the beginning ($t = 0$), the neighborhood set covers at most the whole SOM. Its size diminished with respect to $t$ until the length equals to 3.

After the network is trained through a number of iterations, a converged SOM is generated. The resultant SOM contains after mapping the feature vectors, nodes in the SOM may contain zero or many images. To visualize the similarity

1. Label by similarity: For each node $j$ in the map, find an image $i \in I$ as label such that the Euclidean distance between $x_i$ and $w_j$ is minimal.

$$v_j = \arg \min_{n \in T} \|x_n - w_j\|$$

2. Label by result-mean: For each node $j$ in the map, calculate the mean $y_j$ of weight vectors of images in $R_j$. Next, find the image with minimal Euclidean distance to the mean as label.

$$y_j = \frac{\sum_{i \in R_j} x_i}{|R_j|}$$

3. Label by result-similarity: For each node $j$ in the map, find an image $i \in R_j$ such that the Euclidean distance between $w_j$ and $x_i$ is minimal.

$$v_j = \begin{cases} \arg \min_{n \in R_j} \|x_n - w_j\| & \text{if } |R_j| > 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$ (15)

### 3. Illustration

We implemented the system in Java (JDK 1.1.8) and run it on a Pentium II-233 PC. There are two modules in the system. The first one is feature module, which extract chromatic and textural feature of images in the database. It also includes user-interface for retrieving feature information of each image. A simple query-by-example is implemented in this module for testing the performance of the features. The second one is SOM module which trains SOMs using feature databases. It contains a user-interface to visualize the labeled result for browsing the images. In the following, we will evaluate SOMs using different image features and labeling method.

#### Feature differences

We have used a database with 500 colour images (640x480x24bits) in the following experiments. Figure 5 (a-c) shows a 10x10 SOM trained with different features. Three SOMs are trained for 50 iterations. Learning rate begins at 0.5 and ends at 0.01. Resulting maps are labeled by result-similarity. The number in left-lower corner in each node indicates the number of images in the node ($|R_j|$). After selecting a node in the map by user, the right column displays the images ($R_j$) mapped to the selected node.

All the resulting maps can provide an overview of the database. And more importantly, they contain rich information. They visualize the underlying structure of the feature space by displaying topologically regions of major concepts in the image space and the distribution over the concept regions. We will evaluate the three SOMs in the following.

SOM in figure 5(a) only uses HS-histogram, a chromatic feature. It visualizes the distribution of color of images. For example, images with blues are put on the right-bottom of map. However, it cannot differentiate images with similar colors but different textures. For example, some winter scenery photos and some object photos both are almost black and white, therefore the concepts of these two categorizes of images are not clearly distinguished in the map. Also, objects with different colors like tennis-ball and baseball are separate apart.

---

1 The term “iteration” in our algorithm is defined as number of presentation of all images (see section 2.2). However, most SOM algorithms define “iteration” as number of presentation of input patterns. Using the latter definition, there are 500 x 50 = 25000 iterations in this experiment.
Gabor filter, a textural feature, is used to train the SOM shown in figure 5(b). Contrary to the former SOM, it can separate objects from scenery photos perfectly but photos with similar colors are not grouped together. For instance, photos with blues and greens are in different regions in the map.

The final one combines HS-histogram and Gabor filter. Advantages of chromatic and textural features are found in this map. Objects and scenery photos are separate apart and images with similar colors are close together. Of course the performance is not as good as figure 5(a) in terms of chromatic differentiate performance, and figure 5(b) in terms of textural differentiate performance. Nevertheless, it provides a balance of using both features.

**Labeling differences**

We have also compared different labeling methods. Figures 5(c) and 6(a-b) displays the same SOM trained by same parameters in section 3.1 but labeled with different methods.

The map in figure 6(a) is labeled by similarity, that is, using the most similar image in the database to represent the node. It visualizes best the information of the weights trained in the map thus present an overview of distribution of images in the database and the map is generally very continuous. However, the label image of a node may not be an image mapped to that node. And there are cases of duplicate representative images. If it is used as a user-interface for browsing, users may confuse if their interested image exists repeatedly throughout the map; they do not know which node they should further explore.

Figure 6(b) shows a map labeled by result-mean. The label image is chosen from the most similar image to the mean of features vector of images mapped to that node. It prevents the problem just mentioned of labeling by similar and it finds the best images to represent images mapped to that node. However, the label images are not the most similar one to the weights of the nodes thus the map is visually more fragmented.

The last map is shown in figure 5(c), which is labeled by result-similarity. It has some advantages of labeling by similarity and labeling by result-mean. It provides a balance between representing weights of nodes and representing the features of images mapped to the nodes. Two important properties of result-mean and result-similarity methods are that the label image must be included in images mapped to the node and duplicate label image will not exist.

**4. CONCLUSION**

We have proposed an approach for efficient colour image retrieval. Image feature and SOM are used to visualize a summary of all images in the database. Experiments show that image features can be combined to balance the advantages of different features. And also, SOM labeled by result-nearest is most suitable for representing both the map and the images mapped to the nodes.
The results of experiments are encouraging. We think that SOM is an ideal technique for generating maps for image browsing since it can reflect the relationship among all images in the database. However, there are some restrictions of current approach using SOM. First, if the number of images is large, the number of images mapped in one node will be large. It is inefficient for user to browse. An immediate derived approach to this problem is to generate SOMs for the images mapped to each node. We can apply this approach recursively until the number of images in the node is less than a specific amount. Actually, this approach is similar to the Hierarchical Self-Organizing Map (HSOM) from Zhang et al. and the Multilayered Self-Organizing Feature Map (M-SOM) from Chen et al. [10]. The purpose of these two research are visual indexing and internet categorization respectively. This kind of SOM variations can be applied to our research also. We will have to design appropriate algorithms for labeling.

Another restriction is lack of dynamic maintenance. Insertion or deletion images needs to re-train the SOM. For a normal single layer SOM, we can apply additional training iterations to the existing SOM to adapt the new images. Nevertheless it cannot apply for HSOM/M-SOM approach; modifications of top-level SOM make lower-level SOM to be re-trained. We will continue to study on this field in order to solve this problem.

A problem we have encountered is quantitative evaluation of the results. We claim that the usefulness of maps can be evaluated by practical tests only. Therefore, we will setup human experiment for testing the performance of human in different variables. As SOM provide a visual overview of the database, once subjects have used a SOM for a period, we claim that they will have a better performance due to implicit learning.

Eventually, browsing and searching are not competitive techniques. We will work on a proposal which combines the two major approaches. More features and new combinations of features will be tested. For instance, text-annotation is a useful feature for manual categorization of image database.

5. ACKNOWLEDGEMENT

This project is supported by the Earmarked Grant for Research of the Research Grants Council of Hong Kong, CUHK 7034-98E.

6. REFERENCES


6(a) Label by Similarity
6(b) Label by Result-Mean

Figure 6(a-b): 10x10 SOMs labeled by different methods (both are trained by combined features)