Similarity Image Retrieval System
Using Hierarchical Classification

– Experimental System on Mobile Internet with Cellular Phone –

Masahiro Tada¹, Toshikazu Kato¹, and Isao Shinohara²

¹Department of Industrial and Systems Engineering, Chuo University, 1-13-27, Kasuga,
Bunkyo-ku, Tokyo 112-8551, Japan
{umehara, kato}@indsys.chuo-u.ac.jp
²Kyodo Printing CO.,LTD. 4-14-12 Koishikawa, Bunkyo-ku, Tokyo 112-8501, Japan
i_shinohara@kyodoprinting.co.jp

Abstract. We developed a similarity image retrieval system for database
consists of various kinds and large amount of image data. Our system has the
following features. (1) The similarity retrieval algorithm selects the most
similar group to the visual key step by step as a search space so that the system
could achieve less computation and better precision. (2) The system can easily
adapt each user’s subjective criterion for similarity. (3) The interface is so user
friendly that just showing a key image is enough to invoke content-based image
retrieval. Based on the features above, we developed and evaluated a similarity
retrieval system using cellular phone with digital camera as a multimedia
terminal.

1. Introduction

Most of the image retrieval systems at the major portal sites are designed on text
based matching. Nevertheless text based matching has following problems. (1)
Although text based matching is capable of retrieving candidate images having the
same keywords, it is not capable of ordering them by the similarity of image contents.
Thus, users are obliged to examine a large number of candidates by visual inspection.
(2) Even to the same image, users may subjectively give their own interpretations
based on their experience and knowledge. Therefore, keywords assigned to an image
do not always suit every user. Furthermore, being once created, the keyword index
cannot catch up with changes of users’ interests. (3) There are many kinds of images
which are hardly to describe their contents by simple keywords; such as texture
images, graphic symbols, and design patterns. In order to retrieve such images by
their contents, we have to show a similar instance or a hand-written sketch and to
apply similarity retrieval with evaluating the graphical feature of the images.

Therefore, based on graphical features (GFs) of each image, we have developed a
"Query by Visual Example (QVE)" system for database which consists of various
kinds and large amount of image data. Here, GFs represent the pictorial content of the
image in multi-dimensional vector form.

© Springer-Verlag Berlin Heidelberg 2002
The objectives of our study are following:

(a) Computational modeling of visual perception process based on physiology and psychology: We employ statistical method to model the process in which each user subjectively judges similarity of images by his (or her) own experience and knowledge.

(b) Adapting the model to time-varying criterion of similarity: We develop a statistical supervised learning algorithm which is easy to rearrange the training examples and does not force users to cost much time and patience.

(c) Developing similarity retrieval algorithm: We develop an efficient algorithm in precision and computation for a large-scale database.

In this paper, we propose a modeling method using multivariate analysis for hierarchical classification of database. In this method, when a visual key is shown, the most similar set in hierarchically classified database is selected. Then, only the selected set is regarded as a retrieval domain. This method has achieved better precision and less computation in our experiments.

Since it is too tough to become accustomed to database languages for novice computer users, we also show our prototype system developed for mobile Internet environment. In this system, by using a cellular phone with digital camera (mobile camera) as a database terminal, a user has only to capture key image by mobile camera to retrieve similar images from the database. Thus, a user does not have to know much about database nor computer operations. We have evaluated its performance on our large amount of texture image database.

2. System Architecture

2.1 Content-Based Retrieval Mechanism

The QVE system uses an image itself as a visual key instead of keywords. Visual perception system of human being has nerve circuit which extracts local or global features of brightness and colors in its visual field. By integrating and evaluating these features, we perceive textures and shapes. Therefore, in our experiment, we extract the graphical features (GFs) to simulate the human visual perception in judging similarity. We have designed graphical features in multi-dimensional vector form to represent features perceived from each image (see Chap.3).

The system hierarchically traces the most similar set to the visual key on the bottom level of classification. The set is selected and regarded as a retrieval domain (see Chap.4).

2.2 User Friendly Interface

In Japan, more than 50% people have cellular phones, many of which are equipped with digital camera (mobile camera), and use them as the first aid to access to the Internet instead of personal computer. Cellular phones are very familiar to many people in capturing photos and exchanging them via email. Therefore, we adopt this type of cellular phone as a database terminal for content-based image retrieval.
Fig. 1. Architecture of “Query by Visual Example” system using cellular phone.
The user’s operations for QVE are following:

1. Capture a key image by the mobile camera on cellular phone.
2. Send the image to our server as a visual key by e-mail (cellular phone’s function)
3. Receive the URL by e-mail where the summary report of the retrieved result is uploaded.
4. Visit the URL to access the candidate images.

Our system can be operated without complex database languages. In this paper, we propose this kind of mobile database operation “Retrieve anytime, anywhere, by anybody”.

3. Graphical Features (GFs)

As discussed in Sec.2.1, we should design GFs to simulate human visual perception process which extracts local as well as global features of brightness and colors.

We should also design GFs as shift-invariant and noise-robust, because the camera angle and the lightning conditions are not same even if capturing the same object.

To measure the GFs of textures, local features in neighboring pixels as well as global features defined on whole image plane are important. In our experiment, we adopted 3×3 pixels window to compute local features of directions and curvatures of differentials. Therefore, we adopt relations of neighboring three pixels in the window as local features. Fig.2 shows the local mask patterns, the center of which is the reference point.

![Local mask patterns for GFs](image)

3.1 Shift Invariance and Noise Robustness

It is well known that the autocorrelation function is shift-invariant. Otsu and Kurita proposed its extension to higher orders [3]. The Nth-order autocorrelation functions with N displacement vectors \( \mathbf{a}_1, \ldots, \mathbf{a}_N \in \mathbb{R}^2 \) are defined by

\[
y^n(\mathbf{a}_1, \ldots, \mathbf{a}_N) = \int_{\mathcal{P}} f(\mathbf{r}) f(\mathbf{r} + \mathbf{a}_1) \cdots f(\mathbf{r} + \mathbf{a}_N) d\mathbf{r},
\]

where \( f(\mathbf{r}) \) stands for the image intensity function on the image plane \( \mathcal{P} \), and \( \mathbf{r} \in \mathcal{P} \) is the image coordinate vector. They applied 0th, 1st and 2nd-order autocorrelation functions to face recognition [3] and texture classification [4].

Since the autocorrelation functions include multiplication of noisy intensity values, it is easily affected by noise. Therefore, based on Weber-Fechner’s law, “equal increments of sensation are associated with equal increments of the logarithm of the stimulus,” we defined tri-contrast as follows:
where $a_1$ and $a_2 \in \mathbb{R}^2$ are the displacement vectors (each "*" in Fig. 2 denotes $a_1$ and $a_2$), function $g(r)$ is defined on independent three-dimensional color space which is an extended version of $f(r)$ which is defined on monochrome image, and $r \in P$ is the image coordinate vector (each "+" in Fig. 2 denotes $r$).

Using tri-contrast defined above, we designed local GFs as follows:

$$C^{(t)}(a_1, a_2, r) = \frac{\{g(r + a_1) - g(r)\} + \{g(r + a_2) - g(r)\}}{|g(r + a_1)| + |g(r + a_2)| + 2|g(r)|},$$

where $a_1$ and $a_2 \in \mathbb{R}^2$ are the displacement vectors (each "*" in Fig. 2 denotes $a_1$ and $a_2$), function $g(r)$ is defined on independent three-dimensional color space which is an extended version of $f(r)$ which is defined on monochrome image, and $r \in P$ is the image coordinate vector (each "+" in Fig. 2 denotes $r$).

The denominator and numerator of our tri-contrast denote a power of intensity and a difference of stimulus, respectively. Here, regarding $S$ as intensity of stimulus, tri-contrast correspond to $S / S$. Hence, local GFs designed above correspond to

$$(1 / S) \, ds = \log S.$$

Therefore, our local GFs approximately follow Weber-Fechner's law:

$$I = K \log S + C,$$

where $I$, $K$, $S$, and $C$ denote intensity of sensation, constant, intensity of stimulus and constant, respectively.

Since tri-contrast value is normalized by intensity of stimulus, local GFs are robust to noise. In addition, they are obviously shift-invariant.

Considering the independency of brightness and color, we adopted the L*a*b* color-space. $L^*$, $a^*$ and $b^*$ denote luminosity, red-green chromaticity and yellow-blue chromaticity, respectively. Since $L^*$a*b* color-space consists of three dimensional orthogonal coordinates, each tri-contrast value of $L^*$, $a^*$, $b^*$ is defined independently of the others. Therefore, without considering correlations with $L^*$, $a^*$ and $b^*$, we can compute local GFs regarding only each of the $L^*$, $a^*$, $b^*$ values of reference pixel $r$ as $g(r)$. In addition to these local GFs, we compute the average of each $L^*$, $a^*$, $b^*$ value as global GFs.

### 3.2 Optimal Resolution for GFs

GFs extracted from highest resolution may include only very detail information, while GFs from lower resolution do only very rough information. Since an optimal resolution for classification depends on target image sets, we can not fix the resolution of images beforehand.

A pyramidal image data structure gives a set of images of different resolutions from the highest to lower [5]. By extracting sets of GFs from each of the images in the pyramidal image data structure and selecting the most proper one for each target, an optimal set of GFs for each target are extracted.

[Fig. 3. An example of pyramidal image data structure]
4. Hierarchical Classification

4.1 Advantages of Using Hierarchical Classification

To statistically model the process in which each user subjectively judges similarity of images by his (or her) own experience and knowledge, following requirements should be satisfied.

1. The number of data: To apply reliable statistical supervised learning, we need a large number of training examples.
2. Subjective criterion: We have to give the criterion on judging subjective similarity rule to the examples.
3. Adaptation to time-varying criterion of subjective judgment: Experience changes user's subjective criterion for judging similarity. To adapt the model to this time-varying criterion, users have to repeatedly give a large number of training examples.

To give the criterion, one idea is to give it on judging subjective similarity rule in a similarity matrix form. In showing similarity matrix, each user has to assign the similarity score to each pair of samples. Another idea is to hierarchically classify the database according to subjective similarity. In hierarchical classification, each user subjectively classifies whole images into sets of similar images hierarchically by his (or her) own experience and knowledge. Such hierarchical classification has the following advantages;

1. Compared with a similarity matrix form, non-hierarchical classification is only a rough approximation of subjective similarity, while a hierarchical classification gives rather detailed subjective similarity step by step.
2. In giving criterion, a user may stop the hierarchical classification at any time. The system can apply supervised learning algorithm with the temporarily classified data. A user can continue the classification to give more detailed.
3. Through hierarchical classification, comparison and similarity evaluation of data is rather localized. Therefore, to re-apply supervised learning, or to re-classify training examples, a user does not always have to rearrange the classification of whole training examples. In most cases, a user has only to re-classify the small portion of the database.

Therefore, hierarchical classification is more efficient than non-hierarchical classification.

4.2 Linear Discriminant Analysis (LDA)

An effective set of GF (⊂ GFs) for a classification as well as an optimal resolution for it depends on target image sets and classification levels. Therefore, we separately construct an optimal discriminant space \( \mathbf{z} = (z_1, \ldots, z_N)^T \) for a classification at each level by combinations of GFs \( \mathbf{x} = (x_1, \ldots, x_M)^T \).

Most simple way to combine GFs is to use weighted linear combination

\[ \mathbf{z} = \mathbf{A}^\top \mathbf{x}, \]

where \( \mathbf{A} = [a_{ij}] \) is a weighting matrix. To determine the weights, we apply linear discriminant analysis (LDA).
Given $K$ sets $C_i$ ($i = 1, \ldots, K$) with a priori probabilities $p_i$, the within-class and the between-class covariance matrices of GFs are computed as follows:

$$w = \sum_{i=1}^{K} p_i, \quad v = \sum_{i=1}^{K} p_i \left( \frac{x^{(i)}}{w} - \bar{x} \right) \left( \frac{x^{(i)}}{w} - \bar{x} \right)^T,$$

where $\bar{x}^{(i)}$, $\bar{x}$, and $\Sigma_i$ denote the mean vector of class $C_i$, the total mean vector, and the covariance matrix of class $C_i$, respectively.

Then the optimal weighting matrix $A$ is given by the solution of the following eigen-equations,

$$BA = wA, \quad A^T wA = I,$$

where $B$ is a diagonal matrix of eigenvalues. The $j$-th column of $A$ is the eigenvector corresponding to the $j$-th largest eigenvalue. The dimension of discriminant space $z$ is bound by $\min(K-1, M)$.

### 4.3 Hierarchical Classification Algorithm

Our hierarchical classification algorithm is following.

(a) Regard whole database as a given set. Apply (b) to (d) while $n >> k$, where $n$ and $k$ denotes the number of images in a given set and the dimension number of GFs, respectively.

(b) Classify given set into a small number of subsets.

(c) Select the optimal resolution for the classification of subsets. Apply LDA to construct an optimal discriminant space by linear combinations of GFs.

(d) Regard each subset as a given set.

The number of discriminant spaces made on this algorithm is $\sum_{i=1}^{n_i} n_i$, where $n_i$ and $L$ denote the number of subsets in $i$-th level and the number of levels, respectively.

As discussed in Sec.4.2, the dimension of discriminant space $z$ is bound by the number of target sets. Since the hierarchical classification algorithm of the image data reduces the number of target sets, it provides discriminant spaces of a small dimension.

---

Fig. 4. An example of hierarchically classified database
4.4 Similarity Retrieval Algorithm

First, receiving the visual key, the most similar set in hierarchically classified database is selected step by step, from the top level of classification to the bottoms, using optimal discriminant spaces constructed in Sec.4.3. Secondly, regarding the selected set as a retrieval domain and the optimal discriminant space for the set as an index space, we compute the Euclidean distances between visual key and images in retrieval domain. Then the nearest image from the visual key is regarded as the candidate. By selecting the most similar set to the visual key, dissimilar images by the user’s criterion are excluded from the retrieval domain.

Let us discuss the amount of computation in retrieval. By hierarchical classification, we have only to examine \(O(\log N)\) to select the most similar subset, where \(N\) and \(O(\log N)\) denote the number of whole data and proportional number to \(N\), respectively. In the subset, we have to evaluate each distance for \(O(M)\) times, where, \(M\) denotes the average number of data in each subset on the bottom levels. Even if the number of data grows to \(N'\) (\(N'\gg N\)), the average number of data \(M\) remains in the same order, since users may give more detailed re-classification. Thus, the amount of computation does not increase in proportion to the growth of the database. Therefore, we achieve less computation and better precision in similarity retrieval.

![Similarity retrieval algorithm](image)
5. Experiments

5.1 Hierarchical Classification

As discussed in Chap.4, receiving a visual key, our system traces and selects the most similar set to the key step by step beforehand. Since mistracing may affect similarity retrieval, we need a good recognition ratio of each classification, usually required not less than 85%.

In our experiments, we prepared 5,700 images (240 × 320 pixels). Then, by using hierarchical classification algorithm introduced in Sec.4.3, we classified them into three levels (1st, 2nd, and 3rd level has three, six, and 18 sets, respectively) and constructed the optimal discriminant space for each classification (9 spaces in total). Here, we constructed pyramidal image data structure using reduced images in 1, 1/2, 1/4, and 1/8 resolution. Each set on 1st level has approx. 1,900 images, on 2nd level, each set has approx. 900 images, and on 3rd level, each set has approx. 300 images. When applying LDA to construct the optimal discriminant space, we used half of each set as training examples and the rest as test sets.

To evaluate the efficiency of our GFs and hierarchical classification (HC) method, we have compared recognition ratios on the test sets with each combination; our GFs and color histogram for parameters, HC and non-hierarchical classification (non-HC) methods. Table 1 shows the recognition ratios of each experiment. HC method on our GFs achieved much better recognition ratios than other methods.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Methods</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our GFs</td>
<td>HC</td>
<td>98.0</td>
<td>92.4</td>
<td>85.2</td>
<td>91.3</td>
<td>88.5</td>
<td>87.5</td>
<td>100</td>
<td>89.9</td>
<td>93.0</td>
</tr>
<tr>
<td></td>
<td>Non-HC</td>
<td>70.6</td>
<td>59.4</td>
<td>30.0</td>
<td>57.1</td>
<td>40.6</td>
<td>71.9</td>
<td>77.3</td>
<td>56.3</td>
<td>52.6</td>
</tr>
<tr>
<td>Color</td>
<td>HC</td>
<td>50.6</td>
<td>42.8</td>
<td>44.3</td>
<td>47.0</td>
<td>41.0</td>
<td>40.6</td>
<td>72.2</td>
<td>67.8</td>
<td>67.5</td>
</tr>
<tr>
<td>histogram</td>
<td>Non-HC</td>
<td>42.5</td>
<td>41.9</td>
<td>39.1</td>
<td>42.2</td>
<td>24.9</td>
<td>38.5</td>
<td>65.4</td>
<td>58.9</td>
<td>60.2</td>
</tr>
</tbody>
</table>

5.2 Similarity Retrieval

As shown in Sec.5.1, recognition ratios using HC method on our GFs are good enough. Then, we evaluated the performance of our image retrieval system.

We prepared new 150 images and printed them out. Then, by the operations introduced in Sec.2.2, we captured each of them by the mobile camera, showing it as
a visual key, and invoked similarity retrieval. In our performance evaluation, we measured precision ratio. Precision ratio is defined as follows:

\[
\text{precision ratio} = \frac{\text{the number of relevant candidates retrieved}}{\text{the number of total candidates retrieved}}.
\]

In our experiments, the precision ratio by HC method on our GFs is 81.3%, while those by other methods are less than 40.0%.

From the viewpoint of computation, the index space is only two dimensions and retrieval domain contains approx. 300 images by HC method, while the index space is 17 dimensions and retrieval domain contains 5,700 images by non-HC method.

Therefore, by using HC method, we can achieve better precision and less computation at the same time.

6. Conclusion

We developed hierarchical classification method for similarity retrieval with better precision and less computation. In this mechanism, only a small number of subsets are traced to reduce the index space for similarity retrieval and exclude dissimilar patterns. Thus we achieved both better precision and less computation at the same time even for a large database.

This mechanism, QVE, also provides subjective criteria for content-based image retrieval, which means the system provides more user friendly service with his (or her) kansei model.

We developed our prototype systems for mobile Internet environment. Just capturing an image as a key, a user can retrieve pictorial information from the Internet in a user friendly manner.

References