Learning User Preference in a Personalized CBIR System

Chih-Yi Chiu, Hsin-Chih Lin*2, Shi-Nine Yang
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan 300
* Department of Information Management, Chang Jung Christian University, Tainan, Taiwan 711

Abstract

A new approach for learning user preference in a personalized content-based image retrieval (CBIR) system is proposed in this study. This approach provides users with textual descriptions, visual examples, and relevance feedbacks to find target images. The user query can be expressed by syntactic rules and semantic rules. To build a personalized CBIR system, two problems should be overcome in advance, including the semantic gap and the human perception subjectivity. In this study, the semantic gap is bridged through linguistic term sets, which are represented as fuzzy membership functions. The human perception subjectivity is modelled from relevance feedbacks through profile updating and feature re-weighting algorithms. The user preference is stored in a personal profile for further retrieval. Experimental results support the effectiveness of the proposed approach.

1. Introduction

Content-based image retrieval (CBIR) has received much research interest recently. Although many CBIR systems have been proposed [1-2], there exist several problems that prevent these systems from being commercialized. Two examples of the problems are [2]: (1) the semantic gap between image representations and human perceptions in characterizing an image, and (2) the human perception subjectivity in finding target images. Most CBIR systems provide users with query-by-an-example and/or query-by-a-sketch schemes. Since the features extracted from a query are low-level, it is very difficult for users to supply a suitable example/sketch in the query. To overcome these problems and achieve higher accuracy in a CBIR system, providing users with relevance feedbacks is known to be a promising solution.

Words are convenient and natural ways to express human perceptions. People are used to describe images in words. Therefore, combining words with image examples in a CBIR system becomes an interesting challenge [3-6]. In this study, we propose a new approach for learning user preference in a personalized CBIR system with the help of words. Our system is composed of two major phases, including (1) database creation and (2) query comparison, as shown in Fig. 1. A user can pose textual descriptions and/or visual examples to find target images. If the results are unsatisfactory, the user can feed relevant and/or irrelevant examples for the next retrieval. In our experiments, the precision-recall graph (PR graph) is used to demonstrate the effectiveness of our system.

The rest of this paper is organized as follows. Sections 2 and 3 describe database creation and query comparison, respectively. Section 4 shows the experimental results. Conclusions are given in Section 5.

2. The Database Creation Phase

The purpose of this phase is to bridge the semantic gap between image representations and human perceptions for images. The image is represented as statistical features (Section 2.1), whereas the human perception is expressed as linguistic terms (Section 2.2).
2.1 Feature Extraction

In this study, six Tamura features [7] are used as the image representation. Although the six features characterize low-level statistical properties of textures, they are shown perceptually meaningful. Humans can easily interpret these texture properties by words. It’s the reason that Tamura features are used. For the sake of conciseness, visual properties and computations about Tamura features are not presented here. Detailed discussions can be referred to [7]. After being extracted, each Tamura feature is normalized to the range [0, 1].

2.2 Linguistic Term Generation

We propose a query description language to define a query. The language can be characterized by syntactic rules and semantic rules. The syntactic rules are listed in Table 1; they are context-free and refer to the way that linguistic terms are generated. Degrees of appearance on each Tamura feature can be interpreted as five linguistic terms, as summarized in Table 2.

Table 1. Syntactic rules.

| QueryDescriptionLanguage ::= [QueryExpression ⊕ Connective] |
| QueryExpression ::= <empty> | TextualDescription | VisualExample |
| TextualDescription ::= Negation ⊕ Hedge ⊕ LinguisticTerm |
| VisualExample ::= Negation ⊕ Hedge ⊕ RelevanceAdjective ⊕ TamuraFeature ⊕ #ExampleIDs |
| Negation ::= <empty> | 'not' |
| Hedge ::= <empty> | 'more or less' | 'quite' | 'extremely' |
| LinguisticTerm ::= 'very fine' | 'fine' | 'medium coarse' | 'coarse' | 'very coarse' |
| TamuraFeature ::= 'coarseness' | 'contrast' | 'directionality' |
| 'line-likeness' | 'regularity' | 'roughness' |
| RelevanceAdjective ::= 'relevant' | 'irrelevant' |
| Connective ::= <empty> | 'and' | 'or' |

The semantic rules, as listed in Table 3, refer to the way that the membership function of each linguistic term is generated. According to these rules, the product of two sigmoidal functions is used to generate the membership function of each linguistic term as follows:

Algorithm 1. Membership Function Generation.
Input: Values $x_1, x_2, ..., x_n$ of a Tamura feature for all $n$ texture images in the database.
Output: Membership functions of the five linguistic terms for the feature.

Step 1. Let $c_0 = 0, c_5 = 1$. Initialize five evenly distributed class centers $c_1, c_2, ..., c_5$ of the five linguistic terms on the feature over the discourse [0, 1].

Step 2. Apply the fuzzy c-mean algorithm to adjust the five class centers, i.e., $c_1, c_2, ..., c_5$.

Step 3. Create the membership function $P_j$ of the $j$-th linguistic term (from left to right) on the feature as follows:

$$P_j(x) = \frac{1}{1 + e^{-a(x-b)}} \cdot \frac{1}{1 + e^{-c(x-d)}},$$

where $a = k / (c_j - c_{j-1}), b = (c_j + c_{j-1}) / 2, c = -k / (c_j - c_{j-1}), d = (c_j + c_{j+1}) / 2, k > 0$, and $x$ is the feature value ($0 \leq x \leq 1$).

The parameters of each membership function $(a, b, c, d)$ are recorded in a personal profile to reflect the user preference for image retrieval.

Table 2. Linguistic terms for the six Tamura features.

<table>
<thead>
<tr>
<th>Tamura Feature</th>
<th>Linguistic Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarseness</td>
<td>very fine, fine, medium coarse, coarse, very coarse</td>
</tr>
<tr>
<td>Contrast</td>
<td>very low, low, medium contrast, high, very high</td>
</tr>
<tr>
<td>Directionality</td>
<td>very non-directional, non-directional, medium, directional, very directional</td>
</tr>
<tr>
<td>Line-likeness</td>
<td>very blob-like, blob-like, medium blob-like, line-like, very line-like</td>
</tr>
<tr>
<td>Regularity</td>
<td>very irregular, irregular, medium regular, regular, very regular</td>
</tr>
<tr>
<td>Roughness</td>
<td>very smooth, smooth, medium rough, rough, very rough</td>
</tr>
</tbody>
</table>

Table 3. Semantic rules.

<table>
<thead>
<tr>
<th>Semantic rules for the membership function $\mu_Q$, where $Q$ is a query expression on a Tamura feature:</th>
</tr>
</thead>
<tbody>
<tr>
<td>* LinguisticTerm $\Rightarrow \mu_A(x) = P(x)$, where $P(x)$ is defined in Eq. 1 ($Q$ is a textual description.)</td>
</tr>
<tr>
<td>* #ExampleIDs $\Rightarrow \mu_Q(x) = K(x) = \frac{1}{1 + e^{-a(x-b)}} \cdot \frac{1}{1 + e^{-c(x-d)}},$</td>
</tr>
<tr>
<td>* Hedge $\Rightarrow \mu_H(x) = [\mu_Q(x)]^b$</td>
</tr>
<tr>
<td>* ‘not’ $\Rightarrow \mu_{\neg Q}(x) = 1 - \mu_Q(x)$</td>
</tr>
<tr>
<td>* ‘and’ $\Rightarrow \mu_{\land Q}(x) = \min[\mu_Q(x), \mu_Q(x)]$</td>
</tr>
<tr>
<td>* ‘or’ $\Rightarrow \mu_{\lor Q}(x) = \max[\mu_Q(x), \mu_Q(x)]$</td>
</tr>
</tbody>
</table>

3. The Query Comparison Phase

The goal of this phase is to parse a query (Section 3.1) and to model the human perception subjectivity from relevance feedbacks through profile updating (Section 3.2) and feature re-weighting (Section 3.3) algorithms. New similarity functions will be inferred to reflect the user preference (Section 3.4).
3.1 Query Parsing

A query is defined as a logic combination of query expressions on the six Tamura features. Therefore, it can be parsed by the syntactic rules and then represented as a parse tree, in which a leaf node denotes a query expression and a non-leaf node denotes a connective.

3.2 Profile Updating

The user may mark relevant and/or irrelevant examples for the next retrieval. At each feedback, the personal profile, i.e., the parameters of membership functions, is updated through a gradient descent method. For relevant examples, the previous membership functions are pulled toward to the center of relevant examples. Then the weighted average center of these examples is computed and the error is estimated according to the following error function:

\[ E = [1 - \mu'(x)]^2, \]

where \( \mu' \) is the previous membership function of the feature.

For irrelevant examples, the previous membership functions are pushed away by these examples individually. We define an error function as follows:

\[ E = \sum_j [0 - \mu'(f_j)]^2, \]

where \( f_j \) is the feature value of the \( j \)-th irrelevant example. To minimize \( E \), the gradient descent method is used as follows:

\[ \Delta \phi = -\eta \frac{\partial E}{\partial \phi}, \]

where \( \phi \) is a parameter in \( \mu' \), \( \eta \) is the learning rate, and \( \phi + \Delta \phi \) is the updated parameter stored in the personal profile for further retrieval.

3.3 Feature Re-weighting

When a user starts a new query session, weights of the six Tamura features are assigned to the same value initially. After several feedback iterations, the user’s emphasis on particular features can be predicted from the feedback history. We propose a feature re-weighting algorithm to update the weight on each feature as follows.

Algorithm 2. Feature Re-weighting.

**Input:** A series of previous \( k \) weights, denoted as \( W^{(k)} \), the query expression \( Q \) on a Tamura feature.

**Output:** A series of previous \( k+1 \) weights, i.e., \( W^{(k+1)} \), and the similarity between \( Q \) and \( I \) on the feature, denoted as \( s_Q(I) \).

**Step 1.** If there is no relevant example in \( Q \), set the parameter \( \kappa = 1 \). Otherwise let \( \kappa = \cos(\sigma \times \pi / 2) \), where \( \sigma \) is the standard deviation of relevant examples.

**Step 2.** Update \( W^{(k)} \) to \( W^{(k+1)} \) as follows:

\[ W^{(k+1)}_{k+1} = \alpha K + \sum_{i=1}^{K} \beta^{(i)} \times W^{(k)}_i, \]

where \( \beta^{(i)} \) is a series of decreasing coefficients, each of which denotes the corresponding importance in \( W^{(k)} \), and \( \alpha + \sum \beta^{(i)} = 1 \).

**Step 3.** In the parse tree of the query, two query expressions are combined by a connective \( c \). Let \( I \) denotes the feature value of a database image. The weighted membership function \( s_Q(I) \) is defined as follows:

\[ s_Q(I) = \begin{cases} 1 - \frac{W^{(k+1)}_i}{\sum_i W^{(k+1)}_i} & \text{if } c = '\text{and}' \\ \frac{W^{(k+1)}_i}{\sum_i W^{(k+1)}_i} & \text{if } c = '\text{or}' \end{cases} \]

where \( \mu_Q(I) \) is the membership value of \( Q \) for \( I \).

3.4 Similarity Function Inference

After the personal profile is updated and the feature weights are re-weighted, new similarity functions must be inferred to reflect the user preference. The inference method is discussed as follows. If the previous query on a Tamura feature is empty, the current query on the feature is treated as an initial query. To define the membership function \( \mu_Q(I) \) on the feature, consider the following three types of \( Q \):

**Type 1.** If \( Q = \text{<empty>} \), set \( \mu_Q(I) = 0 \).

**Type 2.** If \( Q \) is a textual description, set

\[ \mu_Q(I) = (-1)^{N+1} [N - P_j] \]

where \( P_j \) is defined in Eq. 1, \( h \) is a hedge. \( N = 1 \) if \( Q \) is a negative expression; else \( N = 0 \).

**Type 3.** \( Q \) is a set of \( n \) visual examples. If there is no relevant example in \( Q \), set \( \mu_Q(I) = 0 \). Otherwise, compute the weighted average center \( x \) and the standard deviation \( \sigma \) on the feature and define the membership function as follows:

\[ \mu_Q(I) = (-1)^{N+1} [N - K^h(I)] \]

where \( K \) is defined in Table 3 and set \( \alpha = k / (\sigma + \delta) \), \( b = x - (\sigma + \delta) \), \( c = -a \), \( d = x - (\sigma + \delta) \), \( \delta > 0 \), and \( k > 0 \).

The parameters of \( \mu_Q \) are stored in the personal profile. The similarity function for the query can be inferred by the logic combination of all membership functions on each feature through min-max compositions.

If the previous query expression on a Tamura feature is not empty, the current query expression on the feature is treated as a user feedback. The gradient descent method used in Sec. 3.2 is applied to modify the previous membership functions. The weighted membership function of a feature is computed using Eq. 2. Again, the similarity function is inferred by the logic combination of all weighted membership functions through min-max compositions.
3.5 Similarity Computation

The similarity between a query and each database image is computed by the inferred similarity function. The system will compute the similarity over the entire database and then display the retrieval results.

4 Experimental Results

Two image databases are used to demonstrate the effectiveness of the proposed approach. The first database contains 1443 texture images selected from Corel Gallery Collection. Fig. 2a shows the retrieval results for the query “very fine ∧ very directional ∧ very regular.” The retrieved images are displayed in a descending similarity order from left to right and top to bottom. Fig. 2b shows the retrieval results if we use the second, fifth, and eighth images (in Fig. 2a) as relevant examples.

The second database is created as follows. First of all, we obtain fifty 512×512 images from MIT VisTex. For each of the 50 images, nine 170×170 non-overlap sub-images are cropped and termed as relevant images. Consequently, we have a database that contains 450 170×170 texture images. To measure the system’s performance, each texture image in the database is served as a query. From its retrieved results, the system selects the relevant images of the query as the feedback for next retrieval. The feedback process repeats three times. Fig. 3a shows the PR graph of a conjunction of all features with query re-weighting. The precision and recall increase in the first feedback is the largest. This fast convergence is a desirable situation. Fig. 3b shows the PR graph of the same queries in Fig. 3a, but without query re-weighting. Obviously, the performance with query re-weighting outperforms that without query re-weighting.

5. Conclusions

A new approach for learning user preference in a personalized CBIR system is proposed in this study. According to our experimental results, the semantic gap can be bridged through linguistic term sets. The human perception subjectivity can be modelled from relevance feedbacks through our profile updating and query re-weighting algorithms. After remedying these problems, our personalized CBIR system can achieve higher accuracy for retrieval. The PR graphs have strongly supported the above-mentioned claims.

References