Image Similarity Computation Using Local Similarity Patterns Generated by Genetic Algorithm

Zoran Stejic\(^1\)  \quad Eduardo M. Iyoda\(^1\)  \quad Yasufumi Takama\(^1,2\)  \quad Kaoru Hirota\(^1\)

\(^1\)Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ward, Yokohama 226-8502, Japan

\(^2\)PREST, Japan Science and Technology Corporation (JST), Tokyo, Japan

E-mail: \{stejic, iyoda, takama, hirota\}@hrt.dis.titech.ac.jp

Abstract—Local Similarity Pattern (LSP) is proposed as a new method for computing image similarity. Similarity of a pair of images is expressed in terms of similarities of the corresponding image regions, obtained by uniform partitioning of the image area. Different from the conventional methods, each region-wise similarity is computed using a different combination of image features (color, shape, and texture). In addition, a method for optimizing LSP, based on genetic algorithm, is proposed, and incorporated in the relevance feedback process, allowing the user to automatically specify LSP-based queries. LSP is evaluated on four conventional methods, and the SIMPLIcity, an advanced image retrieval system, LSP brings between 15% and 24% increase in the average retrieval precision. LSP, allowing comparison of different image regions using different similarity criteria, is more suited for modeling human perception of image similarity than the conventional methods.

I. INTRODUCTION

During the last decade, exponential growth in the amount of digital images on the Internet, in various image collections and databases, has led to the rapid growth of the image retrieval field, and, as a consequence, the development of a number of image retrieval systems [12], [4].

The conventional image retrieval process essentially consists of four steps, generally present in all of the existing systems [12]:
1. **Querying**: the user enters the query image to the system, expressing the user’s information need.
2. **Similarity Computation**: the system computes the similarity between the query image and the database images.
3. **Retrieval**: the system retrieves the database images most similar to the user’s query image and presents them to the user.
4. **Relevance Feedback**: the user evaluates the retrieved images as more or less relevant to the query, whereas, based on that, the system adapts the parameters of the similarity computation method and returns to Step 2 (Similarity Computation).

The retrieval process, for a given query image, finishes at a point when the user is satisfied with the retrieved images.

The focus of this paper is on the Similarity Computation and Relevance Feedback steps:

- **Similarity Computation**: we propose a new method for computing image similarity, based on the idea that distinguishing different objects in the image requires different similarity criteria for each object. Consequently, when comparing images, image regions corresponding to different objects should be compared using different combinations of image features. Therefore, the proposed method expresses the similarity of a pair of images in terms of similarities of the corresponding image regions, and allows different regions to be compared using different similarity criteria.

- **Relevance Feedback**: we propose a genetic algorithm (GA) based approach for optimizing the proposed similarity computation method, and incorporate it in the Relevance Feedback step, taking the burden of the explicit query specification off the user.

The proposed method is called **Local Similarity Pattern** (LSP) method. It addresses the problem of the existing methods for image similarity computation, which do not allow different parts of the image to be compared using different similarity criteria. This, despite the flexibility of the existing methods in choosing the optimal similarity criteria, prevents the specification of complex, however natural for a human, similarity criteria used for image comparison (Section II), resulting in low retrieval precision.

The proposed method is systematically evaluated through the comparison with six conventional methods [13], [1], [7], and the SIMPLIcity [8], [16], an advanced image retrieval system, on four test databases totalling over 2,000 images.

Section II gives background on the conventional image similarity computation methods, while Section III proposes LSP as a new image similarity computation method. Section IV proposes a GA-based approach to optimize the LSP method, and incorporate it in the relevance feedback process. Section V describes the image indexing, as well as the performance evaluation of the proposed method.

II. BACKGROUND ON IMAGE SIMILARITY MODELING

**Human vs. computer similarity perception.** For a human, the similarity of images usually means the similarity of objects appearing in the images. Therefore, most of the existing image retrieval systems model the image similarity at the object level [10], [8], [16], [12].

Research in neuroscience [15] and computer vision [5] confirms that, while different objects are primarily characterized by different visual features, when judging image similarity, human does not always put an equal emphasis on all the features characterizing each object. For example, for a user searching the photos of skies, “cloudy, gray sky” and “clear, blue sky” are similar, no matter how different some of their visual features are.

For a computer, an object appearing in the image reduces to a set of pixels, i.e., an image region, meaning that the
similarity of objects corresponds to the similarity of image regions containing the objects [12]. This is an underlying assumption of all of the image retrieval systems mentioned earlier. As stated in [16], “Region-based retrieval systems attempt to overcome the deficiencies of [...] by representing images at the object level.”

**Image similarity and image features.** For modeling image similarity, conventional retrieval systems, e.g., [13], [1], [7], use low-level image features — like color, shape, and texture — resulting in color similarity, shape similarity, etc. However, as the number and variety of images in a database grows, the discriminative power of each single feature becomes insufficient. Therefore, recent systems, e.g., [16], use combinations of features (e.g., color-shape similarity), with relative importance of each feature either fixed, specified by the user, or interactively inferred by the system through the relevance feedback [4], [12].

![Diagram of image similarity](image)

**Fig. 1.** Model of image similarity used in the existing methods, compared with the proposed LSP

Techniques like relevance feedback allow arbitrarily complex combinations of image features to be used for computing image similarity, while not imposing any burden on the user, regarding the choice of features or their relative importance [10], [2]. However, even in the systems using relevance feedback, once the optimal combination of image features is inferred for a given query image, the same similarity criteria to the whole image area limits the ability of the existing systems to model the human perception of image similarity.

**Assumptions about image similarity.** Summarizing, from the preceding discussion we draw the following assumptions about the image similarity, on which the proposed similarity computation method is based:
- Human perception of image similarity is based on the similarity of objects appearing in the image.
- In order to distinguish different image objects, different similarity criteria are necessary for each object.

### III. LOCAL SIMILARITY PATTERN (LSP)

Starting from the assumptions about image similarity, elaborated in the previous section, we propose the Local Similarity Pattern (LSP) method, which expresses image similarity in terms of similarities of the corresponding image regions, and in addition allows different image regions to be compared using different combinations of image features.

Variables and functions used for the formalization of the LSP method are summarized in Table I and Table II, respectively, and explained in the following.

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VARIABLES USED FOR FORMALIZING THE LSP METHOD</strong></td>
</tr>
<tr>
<td><strong>Set</strong></td>
</tr>
<tr>
<td>images</td>
</tr>
<tr>
<td>regions</td>
</tr>
<tr>
<td>features</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FUNCTIONS USED FOR FORMALIZING THE LSP METHOD</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>feature assignment</td>
</tr>
<tr>
<td>region similarity</td>
</tr>
<tr>
<td>image similarity</td>
</tr>
</tbody>
</table>

**Set of images** I represents the image database. **Set of regions** R contains N × N (= n_R) regions obtained by the uniform partitioning of the image area. **Set of features** F contains image features that are extracted from regions and used for computing region-wise similarity. Feature denotes either a simple image feature, e.g., color or texture, or an arbitrary combination of image features, e.g., color-shape.

**Feature assignment function** F_R, given an image region r, assigns an image feature F_R(r) to that region. **Region similarity function** D_R returns the similarity degree...
Image similarity function $D_I$ returns the similarity degree $D_I(i_1, i_2)$ of a pair of images $i_1$ and $i_2$, with respect to image region $r$, and using image feature $f$. Finally, image similarity function $D_I$ returns the similarity degree $D_I(i_1, i_2)$ of a pair of images $i_1$ and $i_2$, being the arithmetic average of region-wise similarities:

$$D_I(i_1, i_2) = \frac{1}{n_R} \sum_{i=1}^{n_R} D_R(i_1, i_2, r_i, F_R(r_i))$$

(1)

The expression for image similarity function $D_I$ depends on the set of regions $R$ and the feature assignment function $F_R$. Consequently, LSP is defined as a structure containing $R$ and $F_R$:

$$LSP \equiv (R, F_R)$$

(2)

In short, LSP is the assignment of image features to image regions, which are obtained by uniform partitioning of the image area.

Despite the uniform partitioning of the image area, LSP is able to identify even the image regions containing objects of irregular (i.e., non-rectangular) shape. E.g., as Figure 2 illustrates, the “sky” region in the template image is identified by forming a patch of “color” blocks in the upper half of the image, corresponding to the “sky”.

In the ideal case, LSP partitions the image area precisely into regions (segments) containing objects like “sky”, etc. A unique combination of image features characterizes each image region, and, in turn, the object(s) contained within that region, which corresponds to human perception and enables the images containing similar objects to be identified and retrieved.

LSP-based image queries. As described, LSP is proposed for computing similarity between the query image and database images. However, each query image, depending on the context in which it is used, can have many different interpretations, with each interpretation requiring different similarity criteria for retrieving similar images [11]. E.g., the same photo of a landscape showing a lake in front of a mountain can, in one context, mean that the user is looking for photos of lakes, while, in a different context, the user might be looking for photos of mountains.

Without specifying the intended interpretation of a query image, it is not possible to define the image similarity criteria. Therefore, according to our definition, a well-defined query consists of the query image together with the intended interpretation (i.e., the corresponding similarity criteria).

The idea behind LSP, as a structure, is to represent the intended interpretation (and the associated similarity criteria) of the query image in a given context. An LSP-based query we propose consists of the query image, that the user selects, and the LSP, that the system automatically infers based on the user’s relevance feedback. Of course, an experienced user could as well manually specify the LSP.

The essential point is that, in order to specify the information need completely, user’s query must include both the query image and its intended interpretation, i.e., similarity criteria.

LSP implementation issues. LSP method can represent complex image similarity criteria, and using them approximate human perception of image similarity. However, for a user not familiar with the relation of image similarity to image partitioning and image features, manually specifying the optimal LSP, given a query image, is difficult. Furthermore, the number of possible assignments of image features to image regions is huge, making it impossible to try out all the possible LSPs and find the best one.

E.g., uniformly partitioning the image area into $7 \times 7$ regions, and employing six types of image features — color, shape, texture, color-shape, color-texture, and shape-texture — gives $6^{7 \times 7} \approx 1.3 \times 10^{38}$ possible LSPs. Furthermore, the space of LSPs is highly discontinuous, meaning that two LSPs differing in only a few regions can correspond to very different similarity criteria.

The size and high discontinuity of the space of LSPs have motivated us to use a GA for finding optimal LSP. In Section IV, we propose a GA-based approach to solve the problem of assigning image features to image regions in an optimal way.

IV. GA-BASED APPROACH TO OPTIMIZE LSP METHOD

This section proposes a GA-based approach to optimize the LSP method, and incorporate it in the relevance feedback process.

A. Image Retrieval as Optimization Problem

Before GA can be employed to find a LSP which optimizes the retrieval precision, the retrieval process using LSP-based queries must be formalized as an optimization problem. Variables used for the formalization are summarized in Table III and explained in the following.

Set of $I$ represents the image database. Set of $Q$ represents the query images chosen by the user. Set of relevant images $A(q)$ for query $q$ represents the cor-
rect answers to a query (i.e., the ground truth), as defined by the user. This set is initially, at the very beginning of the retrieval process, a singleton, containing only the query image itself, and gradually grows as the user evaluates the retrieved images through the relevance feedback. Set of LSPs \( S \) represents the proposed LSP, according to which the system computes the similarity between the query image and database images. Finally, set of retrieved images \( R(q, s) \) for query \( q \) and with respect to LSP \( s \) represents the images retrieved by the system for a given query.

Based on the preceding definitions, retrieval precision for query \( q \) and LSP \( s \) is defined as:

\[
P(q, s) = \frac{|R(q, s) \cap A(q)|}{n_{A(q)}} \in [0, 1] \tag{3}
\]

representing the ratio of the relevant images that the system retrieved, among the highest-ranked \( n_{A(q)} \) images.

At this point it is necessary to notice that, in response to query \( q \) and using LSP \( s \), the system generates the ranking of all database images (i.e., \( n_{R(q,s)} = n_I \)), according to the similarity to the query image (using the image similarity function \( D_I \), Equation 1). However, only the highest ranked \( n_{A(q)} \) images are considered, since the images most similar to the query image, i.e., top-ranked images, are the most important ones for the user.

Following these definitions, the objective is to maximize the retrieval precision \( P(q, s) \), with respect to LSP \( s \), for a given query \( q \):

\[
\max_{s \in S} P(q, s) \quad \forall q \in Q \tag{4}
\]

Maximizing \( P(q, s) \) with respect to LSP \( s \) means searching for a LSP that, given a query image \( q \), results in the highest ratio of the retrieved relevant images, i.e., best approximates the user’s perception of the image similarity.

**B. GA for Optimizing LSP-based Image Queries**

Given a user’s query image, a genetic algorithm (GA) is used to automatically generate a LSP, and complete a LSP-based query. The objective is to generate a LSP which closely approximates the user’s perception of image similarity, and results in optimal retrieval precision (in a user-defined sense). The underlying mathematical model was presented in Section IV-A.

**Genetic Algorithms.** GA is a domain-independent problem-solving method, that attempts to find a sub-optimal or optimal solution to a problem by genetically breeding the population of individuals, where each individual represents a possible solution to a given problem [6]. Each individual has an associated fitness value, which expresses the quality of the corresponding solution to a problem. Starting with a population of randomly created individuals, GA progressively breeds a population of individuals over a series of generations using natural selection, crossover (recombination), mutation, and other genetic operations. Individuals in the population, i.e., chromosomes, are represented as fixed-length character strings over a given alphabet.

Solving a problem using genetic algorithm requires specification of: (1) chromosome coding, (2) fitness measure, and (3) GA parameters [6].

**GA for generating LSP.** As described in Section III, we propose a GA-based solution to automatically generate a LSP. The task of the GA is to assign image features to the image regions obtained by uniform partitioning of the image area. In that way, GA defines which image features are used when comparing the corresponding regions of a pair of images. Six (combinations of) image features are employed (set \( F \), Section III): color, shape, texture, color-shape, color-texture, and shape-texture (Section V).

**Chromosome coding.** A chromosome represents image area uniformly partitioned into \( 7 \times 7 \) regions \( (n_R = N \times N, \text{with } N = 7) \), is the size of the set of regions \( R \), Section III). Each gene of a chromosome corresponds to one image region, and stores the type of image feature \( \{f \} \) assigned to that region \( (f \in F \), Section III). The length of a chromosome is \( 7 \times 7 = 49 \) genes.

**Fitness measure.** Since chromosome corresponds to a LSP, fitness of a chromosome is expressed as the retrieval precision of the corresponding LSP-based query (Equation 3).

**GA parameters.** We adopted conventional GA parameters [6]. Selection is roulette-wheel. Crossover is uniform, with probability 0.6. Mutation is standard, with probability 0.1. Population size is 50 chromosomes. Length of the evolution is 500 generations.

**V. PERFORMANCE EVALUATION**

This section describes the choice of image features used for computing the image similarity (Section V-A), as well as the systematic performance evaluation of the proposed method (Section V-B).
A. Image indexing

LSP allows arbitrarily many image features and their combinations to be assigned to image regions, and used for image similarity computation (Section III). In principle, the more features are available, the more complex image similarity criteria can be expressed, and the user’s perception of image similarity can be better approximated.

Since most of the image features used in conventional retrieval systems (e.g., shape, texture), don’t have a meaning at the level of a single image pixel, we define a basic unit of image comparison to be an image region, i.e., a rectangular array of image pixels, which is a common approach in image retrieval [4]. Accordingly, each database image is uniformly partitioned into $7 \times 7$ regions (set $R$, Section III), and image features are extracted from each region, making up the image index. The number of regions ($7 \times 7 = 49$) is sufficiently big to ensure that most of the regions in isolation do not correspond to more than a single object shown in the image.

Regarding the image features extracted from regions, we have chosen three most commonly used in image retrieval [12], [4], as well as their combinations: color, shape, texture, color-shape, color-texture, and shape-texture.

Color features are represented by color moments [13], resulting in a 9-D feature vector. Shape features are represented by edge-direction histogram [1], resulting in a 8-D feature vector. Texture features are represented by texture neighborhood [7], resulting in a 8-D feature vector.

The distance between a pair of feature vectors, which expresses region-wise image similarity with respect to a given feature (function $D_R$, Section III), is computed using weighted Euclidean distance [13] for color moments, and city-block distance [4] for edge-direction histogram and texture neighborhood. For a combination of features, e.g., color-shape, the region-wise similarity is computed as the arithmetic average of the individual (i.e., color and shape) similarities.

B. Experiments and Systematic Performance Evaluation

The performance of the proposed method is systematically evaluated on four test databases:

- **Vistex-60 database**, containing 60 texture images, divided into 10 categories, between 2 and 12 images each. Source is [9], directory pub/FLAT/scene128x128/.
- **Vistex-167 database**, containing 167 texture images, divided into 19 categories, between 3 and 20 images each. Source is [9], directory pub/FLAT/128x128/.
- **Corel-900 database**, containing 900 photographs, divided into 8 categories, one with 200 and the rest with 100 photographs. Source is [3].
- **Corel-1000 database**, containing 1,000 photographs, divided into 10 categories, each with 100 photographs [8], [16]. Source is [3].

All the four test databases originate from the well-known image collections, used for the evaluation of the image retrieval systems [16], [8]. Partitioning of images into semantic categories is determined by the creators of each database and reflects the human perception of image similarity.

Vistex-60 and Vistex-167 databases are small in size, but diverse in the content, including images with both artificial and natural motives. Both databases contain a relatively large number of image categories, which is appropriate for testing the image categorization performance. Corel-900 and Corel-1000 databases are medium in size, with each image category containing a large number of images, and a big diversity of images within the category. Corel-900 database, which focuses only on natural scenes (like vegetation and landscapes), represents an easier categorization problem than the Corel-1000 database, which covers a wider range of categories.

As a performance measure, the retrieval precision is used, being the most frequently used measure of the retrieval system performance [4], [12].

Evaluation of the retrieval precision is performed so that each image, in each test database, is used as a query (set $Q$, Table III). For each query image ($q \in Q$), relevant images (set $A(q)$, Table III) are considered to be those, and only those, which belong to the same category as the query image [10], [16]. Based on this, retrieval precision is computed for each query image (Equation 3). Finally, average retrieval precision is computed for each category, as well as for the whole test database.
In the first part of the experiment, the proposed LSP method is compared to six conventional methods based on color, shape, texture, color-shape, color-texture, and shape-texture similarity, respectively. However, since for all test databases, color-texture similarity method performed the best, we report the results only for that method. Average retrieval precision for LSP and color-texture methods, is shown in Table IV and Figure 3 (Corel-900 database).

In the second part of the experiment, LSP method is compared to the SIMPLIcity\(^1\) — an advanced image retrieval system using a wavelet-based feature extraction method, semantics image classification, and integrated region matching technique for image similarity computation [8], [16]. Average retrieval precision for LSP method and the SIMPLIcity system is shown in Table IV and Figure 4 (Corel-1000 database).

<table>
<thead>
<tr>
<th>Database</th>
<th># Images</th>
<th>Color-texture</th>
<th>SIMPLIcity</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vistex-60</td>
<td>60</td>
<td>48</td>
<td>-</td>
<td>63</td>
</tr>
<tr>
<td>Vistex-167</td>
<td>167</td>
<td>41</td>
<td>-</td>
<td>56</td>
</tr>
<tr>
<td>Corel-900</td>
<td>900</td>
<td>52</td>
<td>-</td>
<td>74</td>
</tr>
<tr>
<td>Corel-1000</td>
<td>1,000</td>
<td>43</td>
<td>46</td>
<td>70</td>
</tr>
</tbody>
</table>

As Table IV summarizes, the proposed LSP method brings between 15% and 24% increase in the average retrieval precision, compared with both: (1) the best of the six conventional methods tested, and (2) the SIMPLIcity, an advanced image retrieval system.

The results suggest that the difference in performance between the LSP method and the conventional methods grows, as the size of the database, and the number of relevant images, increase. This makes the proposed method suitable for the retrieval from the large scale databases, like that on the Internet.

**VI. CONCLUSION**

*Local Similarity Pattern (LSP)* is proposed as a new method for computing image similarity. Similarity of a pair of images is expressed in terms of similarities of the corresponding image regions, obtained by uniform partitioning of the image area. Different from the conventional methods, each region-wise similarity is computed using a different combination of image features (color, shape, and texture). In addition, a method for optimizing LSP, based on genetic algorithm, is proposed, and incorporated in the relevance feedback process, allowing the user to automatically specify LSP-based queries.

The proposed method is systematically evaluated on four test databases totalling over 2,000 images. Compared with six conventional methods, and SIMPLIcity, an advanced image retrieval system, LSP brings between 15% and 24% increase in the average retrieval precision.

The proposed LSP method, allowing comparison of different image regions using different similarity criteria, is more suited for modeling human perception of image similarity than the conventional methods.

**REFERENCES**


\(^1\)Authors would like to thank Prof. J. Z. Wang, at the Pennsylvania State University, PA, USA, for providing the test data.