On a relaxation-labeling algorithm for real-time contour-based image similarity retrieval

Paul W.H. Kwan*, Keisuke Kameyama, Kazuo Toraichi

Institute of Information Sciences and Electronics, University of Tsukuba, Ibaraki 305-8573, Japan

Received 2 September 2001; received in revised form 28 August 2002; accepted 14 November 2002

Abstract

In this paper, we propose a relaxation-labeling algorithm for real-time contour-based image similarity retrieval that treats the matching between two images as a consistent labeling problem. To satisfy real-time response, our algorithm works by reducing the size of the labeling problem, thus decreasing the processing required. This is accomplished by adding compatibility constraints on contour segments between the images to reduce the size of the relational network and the order of the compatibility coefficient matrix. Particularly, a relatively strong type constraint based on approximating contour segments by straight line, arc, and smooth curve is introduced. A distance metric, defined using the negative of an objective function maximized by the relaxation labeling processes, is used in computing the similarity ranking. Experiments are conducted on 700 trademark images from the Japan Patent Office for evaluation.

Keywords: Relaxation labeling; Real-time; Contour; Image retrieval; Trademark

1. Introduction

Image similarity retrieval, known commonly as Content-based Image Retrieval (CBIR), is a technique for retrieving images on the basis of automatically derived features such as color, texture, and shape. In the case where multiple objects exist, their spatial locations are also used to facilitate similarity retrieval. Various approaches for CBIR have been reported in the literature by researchers who derived different retrieval methods and similarity metrics based on single or composite features [1–4]. A number of these efforts have resulted in experimental or commercial systems being constructed [5–11]. While directly comparing the effectiveness between different CBIR approaches based on their retrieval results can be difficult as the similarity metrics used are not exactly identical, it is commonly accepted that an important requirement for an approach to be of practical usefulness is that the response be ‘real-time’, in the sense that similar images can be retrieved in matter of several seconds instead of minutes.

On the other hand, relaxation-labeling algorithms have been applied with different degrees of success to a variety of problems found in computer vision and pattern recognition [12] since the seminal work reported in Rosenfeld et al. (1976) [13]. The class of problems for which relaxation labeling is applicable is one where global labeling problems can be represented by networks of local ones. In other words, given a set of possible labels for each node in a relational network, the goal is to assign a single label in the discrete case (or a set of labels in the probabilistic case) to each node, such that the entire network is consistent subject to a system of compatibility constraints that are inherent in the problem specifications. Whereas an initial labeling over the entire network of nodes is rarely consistent but might contain ambiguities, the strength behind relaxation labeling lies in its use of contextual information throughout the ambiguity reduction process. In the case of image similarity retrieval, one could consider a query image \( S \) as the source of the set of objects while a database image \( V \) as the source of the set of labels of the labeling problem.

Among the successes in applying relaxation labeling is a category of problems known collectively as template matching [14]. A specialization in this category of problems is one associated with character recognition [15–19]. In our own research on handwritten Japanese Kanji and Hiragana character recognition using the Electro Technical
Laboratory’s ETL-8 database [20], it was demonstrated that with initial training a recognition rate of 99.20% for 881 Kanji categories and 98.78% for 75 Hiragana categories could be achieved. However, it must be emphasized that earlier efforts in applying relaxation labeling to character recognition had been developed mostly for off-line applications that did not have a real-time response requirement.

Whereas there had been efforts in adapting relaxation-labeling algorithms for explicit parallel processing on dedicated hardware architectures [21–24], our approach is purely algorithmic and is able to take advantage of parallel implementation to further the performance gain if necessary. We note however that in usual image similarity retrieval scenario, availability of special purpose hardware to the general users is not always feasible.

Motivated by the high recognition rate achievable by applying relaxation-labeling algorithms to matching problems such as character recognition, we propose in this paper a relaxation-labeling algorithm for real-time contour-based image similarity retrieval that treats the matching between two images as a consistent labeling problem. To satisfy real-time response, our algorithm works by reducing the size of the labeling problem, thus decreasing the processing required. This is accomplished by adding compatibility constraints on contour segments between the pair of images to reduce the size of the relational network and the order of the compatibility coefficient matrix. Particularly, a relatively strong type constraint based on approximating contour segments by straight line, arc, and smooth curve is introduced. A distance metric, defined using the negative of an objective function maximized by the relaxation labeling processes, is used in computing the similarity ranking. Experiments are conducted on 700 trademark images from the Japan Patent Office for evaluation.

The remaining of this paper is organized as follows. In Section 2, we briefly describe the steps in characterizing contour segments on both the query and a database image that will be used to represent the objects and the labels of our labeling problem. In Section 3, we present our relaxation-labeling algorithm for real-time contour-based image similarity retrieval that is the focus of this paper. For comparison, we discuss the similarities and differences between our matching scheme and another that was described in a recent work that applied relaxation labeling. In Section 4, we discuss experimental results while Section 5 summarizes this paper and ends with concluding remarks.

2. Characterizing image contour segments

There are two major steps in our approach to characterize contour segments on both the query and a database image. First, the set of joint points connecting various contour segments that make up an image is determined by applying a multi-stage joint points extraction method. Second, the type of each contour joint points is decided by selecting a suitable approximating function for the sequence of points between each pair of adjacent joint points. Together with other attributes of each contour segment, this novel segment type will be used in the relaxation-labeling algorithm to represent the set of objects and labels of our labeling problem.

2.1. Extracting the essential joint points

The set of joint points connecting various contour segments that make up an image is determined by applying a multi-stage joint points extraction method. This method was originally developed for a function-font generating system to accurately extract joint points at connections between different types of contour segments in brush-written characters [25]. As this method is not the main focus of our paper, we simply mention its different stages while referring interested readers to the original paper for more details.

Taking as input a 141 × 123 pixels trademark image that has been through color binarization and edge detection, this method automatically determines and labels the set of joint points in three successive stages: (1) extracting obvious joint points based on the computed discrete curvatures along the edges, (2) extracting additional joint points along each contour segment between pairs of obvious joint points by evaluating the analog curvature, and finally (3) removing any redundant joint points. Fig. 1 shows the joint points extracted from a sample trademark image by this method.

From the set of joint points extracted, a procedure described in Section 2.2 determines a suitable function to approximate the contour segment between each pair of adjacent joint points that corresponds to its segment type.

2.2. Approximating contour segments between joint points

The procedure to determine a suitable approximating function for the contour segment between a pair of adjacent joint points is as follow. First, for the contour segment between a pair of joint points that are both labeled as corners of a straight line by the joint points extraction method, it is approximated by a linear function that passes through
the pair of adjacent joint points. Its segment type is a straight line.

Second, for the contour segment between a pair of joint points that are both labeled as corners of an arc, it is approximated by a linear combination of trigonometric functions. Its segment type is an arc.

Finally, for the contour segment between a pair of joint points in which either or both are not labeled, this procedure will first attempt to approximate it by a linear function. Here, the decision is based on the following approximation error criterion, calculated using a pair of parametric functions representing the x and y coordinates as a function of a variable \( t \), respectively.

\[
e = \max_{0 \leq t \leq m-1} (s_x(t_i) - x_i)^2 + (s_y(t_i) - y_i)^2
\]

Here, \( m \) is the number of points on the contour segment. When \( e < 0.90 \) for all \( t_i \), the contour segment is labeled as a straight line. Otherwise, the procedure will try to approximate it as an arc. However, if the approximation error criterion of Eq. (1) is not satisfied, it will be approximated by a smooth quadratic piecewise polynomial. Its segment type is a curve.

In addition to being an attribute for objects and labels of the labeling problem, the segment type is used together with other attributes in defining the compatibility constraints that are instrumental in reducing the size of the labeling problem. Experiments are performed on trademark images with and without adding the novel segment type constraint for comparative evaluation.

3. Relaxation-labeling algorithm for real-time image similarity retrieval

3.1. Overview of algorithm

Relaxation-labeling algorithms, as briefly stated in Section 1, can be considered as parallel and iterative processes developed to solve the so-called (continuous) labeling problem where one has to assign labels to a network of objects so as to satisfy a set of domain-specific constraints [26]. Often, the set of objects \( A \) and the set of labels \( \Lambda \) of a labeling problem are denoted as follows:

\[
A = \{a_1, \ldots, a_n\}, \quad \Lambda = \{\lambda_1, \ldots, \lambda_m\}
\]

Here, \( n \) and \( m \) are the numbers of objects and labels, respectively.

In the case of image similarity retrieval, we are comparing a query image with a database of images to find the best matching configuration between the query and each database image, and to rank their similarity quantitatively by means of a distance metric. As such, we can consider a query image \( S \) as the source of the set of objects and a database image \( V \) as the source of the set of labels of our labeling problem, obtained through a local measurement process applied to both of these images. In this work, the two-step approach in characterizing image contour segments described in Section 2 assumes the role of such a local measurement process.

Whereas the goal of the relaxation-labeling operators as originally described in Rosenfeld et al. (1976) [13] was to obtain a set of consistent labeling assignments with respect to the domain-specific constraints of the problem, it might be difficult to achieve in the case of image similarity retrieval. Here, the objective is not necessarily to find a complete and consistent labeling (where every object has a consistent label assigned) but a best partial labeling, in the sense that as many objects as possible could be assigned a consistent label within the limit of the domain-specific constraints.

As we have shown in Kwan et al. [27], by assigning a special Match-All label \( \lambda_{m+1} \) to objects that are not assigned any candidate labels by an initial pairing process, a consistent labeling for the entire relational network can be obtained when a smaller problem involving only the subset of objects and labels that are paired attains a consistent labeling by the relaxation-labeling algorithm. In other words, \( \lambda_{m+1} \) is considered a consistent label for every ‘unmatched’ object of the original labeling problem. The set of ‘matched’ objects and their most consistent labels ranked by the labeling probabilities constitute a best matching configuration between the pair of images.

In terms of the effects on the dynamics of our relaxation-labeling algorithm, the smaller labeling problem facilitates a reduction in the number of parallel labeling probability-updating processes when executing the algorithm. Furthermore, the order of the compatibility coefficient matrix referenced by these probability-updating processes is reduced due to the smaller number of objects and labels involved in the relational network. Together, these reductions bring about a decrease in the amount of processing required, thus enabling our relaxation-labeling algorithm to satisfy the response requirement of real-time image similarity retrieval.

The remaining of this section is arranged as follows. In Section 3.2, we will explain how to represent the set of objects and labels of our labeling problem by the contour segments obtained from the local measurement process described in Section 2. In Section 3.3, we will show how to reduce the size of our labeling problem by enforcing compatibility constraints on attributes of objects and labels between the query and a database image. In Section 3.4, we will describe the process in finding the most consistent labeling for the objects and labels of our labeling problem by considering compatibility relations among neighboring objects when updating the set of labeling probabilities in the relational network. In Section 3.5, we will define a distance metric in terms of the negative of an objective function that is maximized by our relaxation-labeling algorithm to determine the similarity ranking between the query and the database images. Lastly, in Section 3.6, we compare our
3.2. Objects and labels of the labeling problem

The sets of contour segments obtained from the query and a database image represent the objects and labels of our labeling problem. In turn, each contour segment is represented as a feature vector having attributes that include a pair of start and end points for the segment in \(x\)–\(y\) coordinates, the length of the segment, a pair of start 1/3 and end 1/3 gradients for the segment, two lists of neighboring segments (one for the start and another for the end point), and the segment type.

Whereas the start 1/3 gradient of a contour segment is determined from the direction of the vector connecting its start point and the first ternary point, the end 1/3 gradient is determined from the direction of the vector connecting the second ternary point and its end point. In addition, for a contour segment \(s_i\), the list of ‘start’ neighboring segments is constructed from those segments that have their end points within a certain neighborhood of the start point of \(s_i\). In this paper, without loss of generality, we simply limit the size of the list to have no more than four of its closest neighbors. The list of ‘end’ neighboring segments is obtained likewise.

Altogether, the set of feature vectors corresponding to the contour segments obtained from a trademark image can be represented as follow:

\[
O_i = \{x^{(s)}_i, x^{(e)}_i, y^{(s)}_i, y^{(e)}_i, L_i, \theta^{(s)}_i, \theta^{(e)}_i, T_i, \} \quad \text{for} \quad i = 1, \ldots, n
\]

Here, \(O_i\) is the \(i\)th feature vector, \((x^{(s)}_i, y^{(s)}_i)\) and \((x^{(e)}_i, y^{(e)}_i)\) are the co-ordinates of the start and end points, \(L_i\) is the length of the contour segment in pixels, \(\theta^{(s)}_i\) and \(\theta^{(e)}_i\) are the start and the end 1/3 gradients in radians, \(\{N^{(s)}_{ij}\}_{j=1}^{l_i}, \{N^{(e)}_{ij}\}_{j=1}^{l_i}\) are the lists of ‘start’ and ‘end’ neighboring segments, and \(T_i\) represents its segment type.

As an illustration, Table 1 lists the first 15 out of 29 feature vectors for the contour segments of the trademark image being shown in Fig. 1.

Attributes of these feature vectors are used in defining the compatibility constraints on contour segments between the query and a database image. As explained in Section 3.3, both the number of objects and labels in the relational network and the order of the compatibility coefficient matrix could be reduced by making use of these compatibility constraints.

3.3. Reducing the size of the labeling problem

As stated in the section on the overview, a reduction in the size of the labeling problem is crucial to our relaxation-labeling algorithm in achieving the response required by real-time image similarity retrieval. The approach we adopt in effecting such reduction is a two-step process. First, a reduction in the size of the relational network by pairing objects with labels according to a set of compatibility constraints using attributes of their corresponding contour segments. Those objects that are unpaired will be excluded from the relational network. Second, a reduction in the order of the compatibility coefficient matrix referenced by the relaxation labeling processes is achieved by considering only the subset of objects and labels obtained in the previous step when computing the compatibility coefficients.

In the first step, for every pair of object \(a_i\) and label \(l_k\) in the original labeling problem, the decision to include \(l_k\) as a candidate label for \(a_i\) is based on whether their attributes satisfy the following set of compatibility constraints:

\[
|L_i - L_k| \leq C_1 \times \min(L_i, L_k), \quad |\theta^{(s)}_i - \theta^{(s)}_k| \leq C_2, \quad |\theta^{(e)}_i - \theta^{(e)}_k| \leq C_2, \quad T_i = T_k.
\]
Here, $C_1 = 0.1$ and $C_2 = 15$ are parameters determined from repeated experiments. Depending on the size of the images, these values might vary. Notice that the last constraint involves the comparison of segment types between $a_i$ and $a_k$. This is a relatively strong constraint that serves to limit further the number of candidate labels for each object. For those objects that are unpaired, we assign the Match-all label $\lambda_{a_i+1}$ as explained in Section 3.1 on the overview of this algorithm. The set of objects that are paired, together with their corresponding lists of candidate labels, will be used to construct the actual relational network of our labeling problem.

In the second step, based on the relational network already constructed, our objective is to define a compatibility coefficient matrix of a lower order than the one that would include all the objects and labels in the original labeling problem. Therefore, if the relational network involves $n'$ objects and $m'$ labels, where $n' < n$ and $m' < m$, then the compatibility coefficient matrix can be considered as a four-dimensional matrix $R_{i,j,k,l}^{n',m',i,j,l}$ of real numbers. Each element of this matrix denotes a compatibility coefficient $r_{ij}(k,l)$ that represents the degree of mutual support between two local labeling assignments. That is, object $a_i$ is assigned label $l_k$ when its neighboring object $a_j$ is assigned label $l_l$ simultaneously.

Strictly speaking, in the formulation of our labeling problem, we define two compatibility coefficient matrices instead of one. One of these matrices is associated with the start point’s compatibility relations while the other is concerned with the end point’s compatibility relations. Both of these matrices are continuously referenced by the relaxation labeling processes in updating the set of labeling probabilities of the relational network. Thus, a reduction in the order of these compatibility coefficient matrices has a positive impact on decreasing the amount of space and time required to store and access their individual elements, contributing to an overall reduction in the amount of processing required.

To complete our discussion on the definition of these compatibility coefficient matrices pertaining to the smaller labeling problem, we will define below the rules for evaluating the values of the coefficients for the start point’s compatibility relations. Rules for evaluating the end point’s compatibility coefficients are defined likewise. Assume the sets of objects and labels represented in the relational network be $A$ and $A'$, respectively, where $A' \subseteq A$ and $A' \subseteq A$. For every pair of objects $a_i, a_j \in A'$ in the relational network, together with their corresponding labels $\lambda_k, \lambda_l \in A'$, the compatibility coefficients for their start point’s compatibility relations are defined as:

$$
\mu_{ij}^{(s)}(k,l) = \begin{cases} 
0 & \text{if } a_j \not\in [N^{(s)}_{ij}]_{y=1}^{-1}, \\
\min\{Q, 0\} & \text{if } a_j \in [N^{(s)}_{ij}]_{y=1}^{-1}.
\end{cases}
$$

Here, $Q = \min(1 - W_1 \times |d_{ij} - d_{ij}| + C_3, C_4)$

$$+ \min(1 - W_2 \times |d_{ij} - d_{ij}| + C_5, C_4) - W_3,$$

$$d_{ij} = \sqrt{(x_i^{(s)} - x_j^{(e)})^2 + (y_i^{(e)} - y_j^{(e)})^2},$$

$$d_{kl} = \sqrt{(x_k^{(s)} - x_l^{(e)})^2 + (y_k^{(e)} - y_l^{(e)})^2},$$

$$d_{ik} = d_{ik}^{(s)} - d_{ik}^{(e)}, \quad d_{jl} = d_{jl}^{(s)} - d_{jl}^{(e)}.$$\]

$W_1 = 0.05$ and $W_2 = 0.01$ represent the weights given to the difference in the relative positions and the pointing directions, respectively, while $C_3 = 0.5$, $C_4 = 1.0$, and $C_5 = 0.5$ are parameters determined from repeated experiments to compensate for small differences in measurements. Of particular interest is $W_3 = 0.2$ that is introduced as a penalty for pair-wise ‘type’ incompatibility involving the pair of neighboring objects.

There are two properties of the compatibility coefficients used in this paper that are worth mentioning. First, we have treated $r_{ij}(k,l)$ as the correlation between two simultaneous events that object $a_i$ has the label $\lambda_k$ when $a_j$ has the label $\lambda_l$. This interpretation is similar to the one developed for the nonlinear probabilistic models described in Rosenfeld et al. (1976) [13] that was defined based on the covariance and correlation computed for a pair of events. The difference between our and that of Rosenfeld et al. (1976) is that the values of $r_{ij}(k,l)$ satisfy the condition $0 \leq r_{ij}(k,l) \leq 1$ instead of $-1 \leq r_{ij}(k,l) \leq 1$. This is equivalent to relaxing the contributions from negatively correlated events by setting their compatibility coefficients to zero. We have found from experiments that this relaxed condition has not negatively affected our relaxation-labeling algorithm for real-time image similarity retrieval in terms of convergence to a consistent labeling.

Second, the compatibility coefficients used in this paper are symmetric in the sense that the condition $r_{ij}(k,l) = r_{ij}(l,k)$, for all $i, j, k, l$ is satisfied. As we will discuss next on the labeling probability-updating processes of our algorithm, this symmetry condition and the existence of an objective function that is maximized by the iterative labeling processes are sufficient to guarantee a consistent labeling is attained at convergence by a proved theorem (i.e. Theorem 5.1) in Hummel and Zucker [28]. This is important to our relaxation-labeling algorithm because when it converges, we can ensure that the set of ‘matched’ objects and their most consistent labels together constitute a best matching configuration between the pair of images.

3.4. Finding the most consistent labeling

Finding the most consistent labeling for the set of objects and labels in the relational network is the key to obtaining a best matching configuration between the pair of images.
Starting with the set of initial labeling probabilities that characterizes the state of the relational network, the goal is to update these probabilities iteratively via a probability-updating scheme in order to increase the overall consistency of the relational network.

In our algorithm, the set of initial labeling probabilities are computed after the pairing of objects with labels described in Section 3.3 has been completed. For every object \( a_i \in A' \), \( A_i \) represents the set of candidate labels determined by this pairing process. Then, for each label \( \lambda_k \in A_i \), the following definition determines the corresponding labeling probability:

\[
p_i^{(0)}(k) = \frac{p_i(k)}{\sum_{k'} p_i'(k')}, \tag{7}
\]

\[
p_i'(k) = \max\{1 - W_k \times \max(|L_i - L_k| - C, 0), 0\}.
\]

Here, \( W_k \) is the weight assigned to the difference in length between the contour segments of object \( a_i \) and the label \( \lambda_k \) in reducing their initial labeling probability. In addition, \( C \) is an offset for compensating small differences in length caused by signal noises appearing on the images.

Once the set of initial labeling probabilities are computed, their values are iteratively updated by the relaxation-labeling algorithm via the following probability-updating scheme, by taking into account the compatibility coefficients for both the start and the end point’s compatibility relations.

\[
p_{t+1}(l) = \max_p \left[ 1 + q_{dik} \right] \times \left[ 1 + q_{eik'} \right],
\]

\[
q_{dik} = \sum_{j} L_j \times L_i, \quad q_{eik'} = \sum_{j'} L_{i'} \times L_i,
\]

\[
q_{dik} = \sum_j L_j \times \max\left( r_{dij}^{ij}(k, l) \times p_{ji}^{(l)} \right),
\]

\[
q_{eik'} = \sum_{j'} L_{i'} \times \max\left( r_{dj}^{ij}(k, l') \times p_{ji}^{(l')} \right).
\]

Here, the indices \( j \) and \( j' \) denote the start and the end point’s neighboring objects for object \( a_i \) in the relational network, respectively. Moreover, the quantity \( q_{dik} \) represents the degree of support given to the assignment of \( \lambda_k \) to \( a_i \) according to the combined evidences of neighboring labeling probabilities and compatibility relations.

The probability-updating scheme defined here resembles the nonlinear probability-updating operator for nonlinear probabilistic models in Rosenfeld et al. (1976) [13]. It has these useful properties. First, the denominator serves to guarantee that the labeling probabilities \( p_{iik}'s \) continue to sum to 1. Second, it ensures that the denominator will be non-zero in the extreme case that there is no support given to the assignment of \( \lambda_k \) to \( a_i \) from its neighbors. Third, a large value of \( q_{dik} \) at time \( t \) increases the value of \( p_{iik}^{(t+1)} \), while a small value contributes to little change, underlining the monotonic property of this probability-updating scheme that is useful in speeding up convergence by encouraging an early winner.

So far, we have not discussed the convergence of our relaxation-labeling algorithm in relation to this probability-updating scheme. Here, rather than proving its convergence formally, we will show that our algorithm converges to a consistent labeling through satisfying the conditions of a theorem on the notion of consistency and convergence that was proved in the foundation paper of Hummel and Zucker [28]. In the original paper, this theorem (that is Theorem 5.1) was stated as follow:

Suppose that the matrix of compatibilities \( \{ r_{ij}(\lambda, \lambda') \} \) is symmetric, i.e.,

\[
r_{ij}(\lambda, \lambda') = r_{ji}(\lambda', \lambda)
\]

If \( A(\tilde{p}) \) attains a local (relative) maximum at \( \tilde{p} \in K \), then \( \tilde{p} \) is a consistent labeling.

\( A(\tilde{p}) \) is what Hummel and Zucker referred to as the average local consistency. It is a function of a matrix \( \tilde{p} \) of weighted labeling assignments on the set of objects and labels in the labeling network. These weighted labeling assignments play a similar role as the labeling probabilities defined in this paper. Its definition was given as

\[
A(\tilde{p}) = \sum_{i=1}^{n} \sum_{\lambda} p_i(\lambda) s_i(\lambda) \tag{9}
\]

Here, \( s_i(\lambda) \) is the support function for label \( \lambda \) on object \( a_i \), defined in terms of the compatibility coefficients weighted by the labeling weight \( p_i(\lambda) \), as follow:

\[
s_i(\lambda) = s_i(\lambda; \tilde{p}) = \sum_{j=1}^{n} \sum_{\lambda'} r_{ij}(\lambda, \lambda') p_j(\lambda') \tag{10}
\]

Substituting Eq. (10) into Eq. (9), we have

\[
A(\tilde{p}) = \sum_{i,\lambda} \sum_{j,\lambda'} r_{ij}(\lambda, \lambda') p_i(\lambda) p_j(\lambda') \tag{11}
\]

where \( A(\tilde{p}) \) is expressed as the summation of quadratic forms involving the labeling weights.

Notice that \( A(\tilde{p}) \) is a scalar quantity. In Hummel and Zucker, it acted as an objective function that was maximized to guide the transition from the initial labeling to the local (relative) maximum. When this is attained, \( \tilde{p} \) is a consistent labeling according to the theorem.

In the formulation of our relaxation-labeling algorithm, an objective function analogous to \( A(\tilde{p}) \) exists. We denote
this objective function as $A'(\bar{p})$, which is defined as follow:

$$A'(\bar{p}) = \sum_{ij} R_{ik}^{j}$$

where

$$R_{ik}^{j} = p_i(k)$$

Here, $k$, $j$, and $l$ are indices of the most consistent labels for objects $a_i$, $a_j$, and $a_l$, respectively. Here, $j$ and $j'$ denote the start and the end point’s neighboring objects, respectively. $A'(\bar{p})$ is written as the sum of the $R_{ik}^{j}$'s that are quadratic terms of the labeling probabilities $p_i(k)$'s, resembling the $A(p)$ defined in Hummel and Zucker.

The existence of $A'(\bar{p})$, together with the symmetry requirement on the compatibility coefficients satisfied, ensure that if our relaxation-labeling algorithm attains a local (relative) maximum on $\bar{p}$, then $\bar{p}$ is a consistent labeling by the result of Theorem 5.1 in Hummel and Zucker. In other words, those ‘matched’ objects and their most consistent labels together constitute a best matching configuration between the pair of images.

### 3.5. Distance metric for similarity ranking

In defining the metric to measure the distance between a pair of images, we make use of the fact that when $A'(\bar{p})$ is maximized at convergence, our relaxation-labeling algorithm has achieved the most consistent labeling for the set of objects and labels in the relational network. In other words, the pair of images is at their closest. Since $A'(\bar{p})$ is a maximized value, we can define the distance metric in terms of its negation, evaluated after the algorithm has converged. From the set of distances computed between the query and each database image, a final similarity ranking can be determined.

Using the set of quadratic terms $R_{ik}^{j}$ defined in Eq. (12), the distance metric $D$ used for computing the distance between the query and a database image is defined as the following scalar quantity:

$$D = \sum_i \left( 1 - R_{ik}^{j} \right) \times L_i + \sum_j L_{ij} + \sum_{k'} L_{ik'}$$

(13)

Here, $L_{ij}$ and $L_{ik'}$ are the length of the corresponding contour segments for an ‘unmatched’ object $a_i$ and an ‘unpaired’ label $A_{ik'}$ of the query and the database image that have been excluded from the relational network after the initial pairing process. These summations involving $L_{ij}$ and $L_{ik'}$ in the numerator are included as penalty for having a large number of unmatched contour segments in the two images.

### 3.6. Related work

In this section, we compare our matching scheme and another that was presented in Pelillo et al. (1999) [29] that made use of relaxation labeling. In Ref. [29], the authors applied their scheme on matching hierarchical structures using association graphs to matching shock trees, which is the abstraction they chose for representing shapes or simple closed planar curves. Shocks, as the authors explained, are singularities formed during a curve evolution process, which is inspired by Blum’s classic work on axis-morphologies for defining equivalence classes of objects [30].

In their experiments, both plain and attributed shock trees were compared. The attributes are geometric information contained in each shock sequence (i.e. the location, time of formation, speed, and direction of each shock). The authors showed by experimental results that this led to better discrimination between shapes than that provided by shock tree topologies alone.

To summarize, the followings are the list of similarities and the list of differences between the two schemes.

#### (1) Similarities

- Both schemes operate on relational structures that are derived from the shapes being matched;
- The matching algorithms adopted in both of these schemes are relaxation labeling based;
- Both matching algorithms involve finding local (global) maximums of an ‘energy’ function of their respective dynamical systems.

#### (2) Differences

- The relational structures used in our scheme are simple connected graphs with a flat structure, while those of their scheme are trees (i.e. with hierarchy).
- In our scheme, there is an initial pairing process on the two relational structures using compatibility constraints in order to reduce their sizes before the relaxation labeling processes commence. In their scheme, an attributed association graph is derived from the two attributed shock trees, which is referenced by the relaxation labeling processes.
- In our scheme, a compatibility coefficient matrix is involved in the definition of the quadratic ‘energy’ function while in their scheme, the adjacency matrix of the association graph is used instead.
- In our scheme, the similarity measure is defined using the negation of the ‘energy’ function that is maximized. In their scheme, a score is computed based on
the similarity of the nodes included in the maximum clique found in the association graph.

Finally, it is not clear from their experimental evaluation whether multiple non-contiguous regions could be handled with satisfaction by their matching scheme due to the number of permutations of the sub-trees that could be induced in the process of constructing a ‘parent tree’ in their representation.

On the other hand, in our scheme the experimental results shown particularly in Table 3, indicates that this kind of situations can be handled.

4. Experimental results and discussion

Experiments are performed on a database of 700 randomly selected trademark images (each $141 \times 123$ pixels) obtained from the Japan Patent Office’s website for performance evaluation. Ten of them were selected as test images. To verify the effectiveness of our algorithm for image similarity retrieval, several types of ‘noises’ were introduced to create variants of the test images. These include rotating, shearing, pixel shifting, shrinking, and adding/removing joint points from the images.

On a Pentium II 400 MHz PC with 64 MB memory, the ten most similar images of each test image are retrieved for evaluation. Tables 2 and 3 show the results from two of the test images. A choice of using or ‘not using’ the segment type is allowed. In Table 2, the ranking is based on ‘not using’ the type constraint while in Table 3, the ranking is based on using it. The symbol ‘–’ indicates the particular image is not ranked within the top ten result.

We observe that by using the type constraint, images became farther apart as indicated by the distances computed in Tables 2 and 3. In terms of the last three rows of results, they are meant for demonstrating the magnitude of reduction in the size of the labeling problem. The number of ‘non-corresponded’ segments is equal to the sum of the unmatched objects and labels in the original labeling problem. Both the ‘size of labeling problem’ and the ‘size of compatibility matrix’ are ratios of the reduced labeling problem to the original expressed in percentages. Whereas the ‘size of labeling problem’ is computed in terms of the product of numbers of objects and labels, the ‘size of compatibility matrix’ is calculated as the product of their squares due to the four-dimensional compatibility matrix.

From the figures obtained, we are able to make these observations. First, the query image and its variants consistently rank within the ten most similar images returned by our algorithm, thus affirming the effectiveness of our algorithm for image similarity retrieval. Second, excluding the query image itself, the size of the reduced labeling problem and its compatibility coefficient matrix are largely below 20 and 10% of the original labeling problem for the 1st test image, while they are largely below 50 and
30% for the second test image. As we go beyond the ten most similar images, the reductions are expected to be more significant.

Finally, to demonstrate that our relaxation-labeling algorithm for contour-based image similarity retrieval can achieve real-time response, we present in Table 4 the total time required to retrieve the ten most similar images for each of the ten test images. Column 2 reveals the figures when the segment type is included, while column 3 does not. Using the segment type constraint, the average retrieval time is 4.549 s with a standard deviation of 1.101 s. Without using the segment type constraint, the average is increased to 7.305 s with a standard deviation of 3.871 s. By dividing the database of images into roughly even-sized groups and matching against the query on multiple machines in parallel, an even shorter time to obtain the final similarity ranking can be achieved [31].

### 5. Conclusions

Although relaxation-labeling algorithms are considered as computational expensive, and only suitable for off-line applications, we have presented and demonstrated in this paper a relaxation-labeling algorithm for contour-based image similarity retrieval that can satisfy the real-time response. Our algorithm works by reducing the size of the labeling problem, thus decreasing the processing required. This is accomplished by adding compatibility constraints on contour segments between the images to reduce the size of the relational network and the order of the compatibility coefficient matrix. Particularly, a relatively strong type constraint based on approximating contour segments by straight line, arc, and smooth curve is introduced.

We have chosen to demonstrate the effectiveness and efficiency of our algorithm on retrieving similar trademark images because this represents a genuine yet unmet need in government agencies administering trademark registrations for detecting potential infringements.
Moreover, trademark-related information that is disclosed on these agencies’ websites are restricted to keywords search while retrieval based on similarity is not yet available. The applicability of our algorithm, however, is not limited to trademark images and can be extended to other applications having the need for image similarity matching.

Future directions include researching a robust contour smoothing method to remove unwanted noises along image object contours so that the number of contour segments will not become excessive. The number of extracted contour segments affects the size of the labeling problem and thus its computation requirement. Another direction is applying our algorithm as the component responsible for fine matching in a hybrid coarse-to-fine image similarity retrieval scheme.

Most of these hybrid schemes reported in the literature rely on computational expensive methods to do the fine matching such as deformable templates that sometimes more than offset the savings gained by the coarse matching stage.

References