Deformed trademark retrieval based on 2D pseudo-hidden Markov model∗

Min-Ta Chang, Shu-Yuan Chen*

Department of Computer Engineering and Science, Yuan-Ze University, 135 Yuan-Tung Rd., Nei-Li, Chung-Li, Taoyuan, 32026, Taiwan

Received 29 October 1998; received in revised form 17 February 2000; accepted 17 February 2000

Abstract

A new deformed trademark retrieval method based on two-dimensional pseudo-hidden Markov model (2D PHMM) is proposed in this paper. Most trademark retrieval systems focus on color features, shape silhouettes, or the combination of color and shape. However, these approaches adopted individual silhouettes as shape features, leading to the following two crucial problems. First, most trademarks have various numbers of decomposed components, while the silhouette-based approaches cannot handle the variety correctly. Second, the infringement cases in which trademarks are changed by non-rigid deformation, in particular nonlinear deformation, may escape detection. Thus, our method focuses on the overall appearance of trademarks and incorporates color and shape features into 2D PHMM to tackle the above two problems. The reason to involve 2D PHMM is that it has high tolerance to noise and distortion, moreover, contextual information can be incorporated into it in a natural and elegant way. However, 2D PHMM is computation intensive and sensitive to rotation, scale and translation variations. Thus, it is the main originality of this paper to include the advantages of 2D PHMM but to exclude its disadvantages. As a result, similar trademarks can be retrieved effectively, even those with different numbers of components or non-rigid deformation. Various experiments have been conducted on a trademark database to prove the effectiveness and practicability of the proposed method. © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Deformed trademark; Similarity measure; Color; Shape; Color quantization; Log-polar mapping; Invariance; 2D hidden Markov model

1. Introduction

Trademark recognition is an important research issue since the increasing number of registered trademarks puts a heavy burden on manual examiners. Hence, it is imperative to develop similar trademark retrieval for automatic trademark recognition, which in turn speeds up manual examination process.

The developed trademark recognition methods in the literature include those using color features [1], shape features [2–5], or both of them [6–9]. However, most shape features used in these approaches are restricted to silhouettes, thus resulting in two crucial problems related to composition and deformation resemblances. The composition resemblance means that two images may look alike although the numbers of components included in the images may be different. Two examples are shown between Figs. 1(a) and (b). The deformation resemblance can be defined in the similar way except that the difference between two images is caused by deformation. Six examples are shown between Figs. 1(a) and (c), (d) and (f). Past approaches might derive low similarity values between the images having either the composition or deformation resemblances due to the weaknesses listed below.
First, although some approaches concerned more than one silhouettes, the techniques used to handle differences in the numbers of components such as hierarchical representation [2], combination summation [3,4,6–8], histogram intersection method [9] or maximum and average terms of similarities [5] are too rough to tackle the composition resemblance problem correctly. Second, most approaches took into account mild non-rigid transformation rather than the general non-rigid and nonlinear deformation.

Our method copes with the two problems by focusing on overall appearance rather than individual silhouettes. More specifically, we incorporate color and shape features into 2D PHMM to tackle the problems. The hidden Markov model is a popular stochastic method used for attacking problems such as matching observed signal patterns [10,11], character recognition [12] and color image retrieval [13]. The retrieval method in Ref. [13] is effective and flexible. Moreover, it can handle the varieties of images by 2D PHMM, thus its matching performance is better than that of template matching. However, 2D PHMM is computation intensive and sensitive to rotation, scale, and translation variations. Thus, it is the main originality of the proposed method to include the advantages of 2D PHMM but to exclude its disadvantages. First, to reduce computation time, a color quantization is proposed to reduce the computation load as required by true color processing. The log-polar mapping is then employed to reduce the spatial resolution. More important, the log-polar mapping [14–16] can also be used to achieve the rotation, scale and translation invariant requirement. The invariant property together with the flexibility of 2D PHMM lead to the result that similar trademarks can be retrieved effectively, even those with different numbers of components or non-rigid deformation.

More specifically, the proposed method includes two stages: database creation and query comparison. The former is concerned with construction of 2D PHMM for each image in the database. The latter deals with matching the query image to each of the 2D PHMMs in the database.

The database creation stage consists of two major phases: feature extraction and construction of 2D PHMM. The feature extraction phase is further composed of two steps: color quantization and log-polar mapping. A color quantization method based on the YIQ color model is proposed to reduce the number of colors. The quantized colors are defined as cellular decomposition of chromaticity terms followed by luminance decomposition. Accordingly, the number of colors can be reduced from $256^3$ true colors to 36 colors. After color quantization, connected components can be labeled [17] and foreground and background components can be found out. The centroid of all foreground objects is considered as the origin of the polar coordinate system. The log-polar mapping can then be performed on the polar coordinate system to resample the image. The sizes of resampled images can be reduced 18–300 times. More important, the resampled images are rotation, scale and translation invariant which in turn can be used to construct geometric invariant 2D PHMMs.

In the query comparison step, it also consists of two major phases: feature extraction as well as matching between the query image and 2D PHMMs in the database. The first step is the same as that in the database creation step. To match the query image, a modified 2D Viterbi algorithm [12,13] is used to estimate the probability values between the query image and 2D PHMMs in the database. The probability values can be considered matching scores to retrieve similar images. In other words, the images of which 2D PHMMs best match the query can be retrieved.

The remainder of this study is organized as follows. Feature extraction and retrieval method are described in
Sections 2 and 3, respectively. Finally, experimental results and conclusions are given in Sections 4 and 5, respectively.

2. Feature extraction

In this section, we explain how to extract necessary information for constructing 2D PHMM and query comparison. The feature extraction phase is composed of two steps: color quantization and log-polar mapping.

2.1. Color quantization

In this section, color space used in this paper is first defined. Quantized color clusters are then described, followed by the quantization method.

In computer graphics, the RGB model is a popular hardware-oriented color model. However, it is not suitable for color segmentation since its three spectral components are mutually dependent. On the contrary, the YIQ model is more related to human visual system since its luminance and chromaticity signals are isolated. In addition, RGB-to-YIQ transformation is a simple linear transformation. Thus, the YIQ color model was adopted in this paper.

As shown in Fig. 2, the RGB space can be considered a RGB parallelepiped in the YIQ space where solid lines represent visible edges and dash lines represent invisible edges. Note that the RGB parallelepiped has eight corners labeled as Red, Magenta, Blue, Cyan, Green, Yellow, Black and White. The projection of the RGB parallelepiped along the Y-axis onto the IQ plane is a chromaticity hexagon. Assume that the projection for each corner C of the RGB parallelepiped is denoted as C'. The chromaticity hexagon can be denoted by six vertices, Red', Magenta', Blue', Cyan', Green', Yellow'. Note that both White and Black are projected onto the origin of the YIQ model since the Y-axis is just the principal diagonal from White to Black.

The quantized color clusters are defined by two-stage decomposition. First, cellular decomposition is used to divide the chromaticity hexagon into several cells. Each cell is then further divided along Y-axis into final color clusters for luminance quantization. Since trademark images usually include few colors, only 36 representative colors are defined and used in this paper.

The chromaticity hexagon is first divided into 19 Voronoi cells, each of which has special meaning. As shown in Fig. 3, the 19 cells can be classified into three rings according to their saturation values. The inner ring (ring 0) includes only cell 0 which has center at the origin of the YIQ model and represents gray value. The outer ring (ring 2) includes cells 7–18. The centers of the cells 7, 9, 11, 13, 15, and 17 are exactly the six corners of the chromaticity hexagon and represent the respective primary colors. While the centers of the cells 8, 10, 12, 14, 16 and 18 are the middle points of the line segments connected by two neighboring corners of the chromaticity hexagon, respectively. The middle ring (ring 1) includes cells 1–6 whose centers are, respectively, the middle points of the line segments connecting the corners of the chromaticity hexagon and the origin of the YIQ model. According to the geometry constraints described above, the chromaticity values (I, Q values) of the respective cell centers can be easily derived as listed in Table 1. The 19 cells with their cell indexes in respective rings are shown in Fig. 3.

For luminance quantization, each cell will be further divided into clusters on the basis of luminance interval as defined below. For each point P' on the IQ plane, the line through the point P' and parallel to the Y-axis can intersect with the RGB parallelepiped at two extremes, say the bottom extreme $P_b$ and the top extreme $P_t$, as shown in Fig. 2. Thus, for each point P' on the chromaticity hexagon, its luminance interval is $(P_b, P_t)$.

In general, for each cell center, the length of its luminance interval is proportional to the distance from itself to the origin of the YIQ model since the transformation between RGB to YIQ is a linear mapping. Thus, for the cells in the same ring, their centers have the same length of luminance interval but may have different intervals. Moreover, the closer the cell center to the origin, the longer its luminance interval. Accordingly, the cells in the rings 0, 1 and 2 will be, respectively, divided
The initial values of the 36 quantized color clusters

<table>
<thead>
<tr>
<th>Ring index</th>
<th>Cell index</th>
<th>Cluster index</th>
<th>Y</th>
<th>( (I, Q) )</th>
<th>( (R, G, B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>235</td>
<td>(0, 0)</td>
<td>(235, 235, 235)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>194</td>
<td>(194, 194, 194)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>153</td>
<td>(153, 153, 153)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>112</td>
<td>(112, 112, 112)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>71</td>
<td>(71, 71, 71)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>25</td>
<td>(25, 25, 25)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>146</td>
<td>(76, 27)</td>
<td>(235, 108, 108)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>105</td>
<td>(194, 67, 67)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>64</td>
<td>(153, 26, 26)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>160</td>
<td>(35, 66)</td>
<td>(234, 108, 234)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>119</td>
<td>(193, 67, 193)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>78</td>
<td>(152, 26, 152)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>121</td>
<td>(41, 40)</td>
<td>(107, 106, 235)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>80</td>
<td>(66, 65, 194)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>39</td>
<td>(25, 24, 153)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>197</td>
<td>(−76, −27)</td>
<td>(108, 235, 235)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>156</td>
<td>(67, 194, 194)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>115</td>
<td>(26, 153, 153)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>183</td>
<td>(−35, −66)</td>
<td>(109, 235, 109)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>142</td>
<td>(68, 194, 68)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>101</td>
<td>(27, 153, 27)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>220</td>
<td>(41, −39)</td>
<td>(235, 234, 108)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>179</td>
<td>(194, 193, 67)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>138</td>
<td>(153, 152, 26)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>76</td>
<td>(152, 54)</td>
<td>(255, 0, 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>91</td>
<td>(111, 94)</td>
<td>(255, 0, 128)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>105</td>
<td>(70, 133)</td>
<td>(254, 0, 254)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>67</td>
<td>(−6, 106)</td>
<td>(127, 0, 254)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>28</td>
<td>29</td>
<td>(−82, 79)</td>
<td>(0, 0, 255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>104</td>
<td>(−117, 12)</td>
<td>(0, 128, 254)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>179</td>
<td>(−152, −54)</td>
<td>(0, 255, 255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>164</td>
<td>(−111, −94)</td>
<td>(0, 255, 127)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>150</td>
<td>(−70, −133)</td>
<td>(1, 255, 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>188</td>
<td>(6, −106)</td>
<td>(128, 255, 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>226</td>
<td>(82, −79)</td>
<td>(255, 255, 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>152</td>
<td>(117, −13)</td>
<td>(255, 129, 0)</td>
<td></td>
</tr>
</tbody>
</table>

After the image has been quantized, we can obtain the connect-components by common component labeling algorithm [17]. Since the foreground objects must be discriminated from background, we can identify the background as the object whose boundary box is closer to the boundary box of the whole image.

### 2.2. Log-polar mapping

In 1866, Schultz observed that light receptors on the retina are not uniformly distributed [14]. The size and spacing of receptors increase as a function of eccentricity. This idea was introduced on the pattern recognition by the log-polar mapping [14–16]. Moreover, the translation, rotation and scale invariance of the log-polar mapping can be achieved after proper normalization [15]. Thus, we employ log-polar mapping to represent the non-uniform structure lying behind the retina in this paper.

Since the log-polar distribution is based on the polar coordinate system, the origin point and polar axis must be defined. However, how to determine the origin point and polar axis for a polar system is a major problem since the similarity between images can be calculated correctly only if all the similar images are translated to the same origin point and rotated to the same polar axis. The origin point could be estimated by the center of gravity of the distances between each pixel and the 36 cluster centers can be first computed. The pixel is then classified into a quantized color cluster that has the smallest distance to the pixel. After that, the center of each color cluster will be replaced by the mean value of all the pixels assigned to the cluster. Finally, each pixel is colored by the mean value of the corresponding cluster.

After the image has been quantized, we can obtain the connect-components by common component labeling algorithm [17]. Since the foreground objects must be discriminated from background, we can identify the background as the object whose boundary box is closer to the boundary box of the whole image.

### 2.2. Log-polar mapping

In 1866, Schultz observed that light receptors on the retina are not uniformly distributed [14]. The size and spacing of receptors increase as a function of eccentricity. This idea was introduced on the pattern recognition by the log-polar mapping [14–16]. Moreover, the translation, rotation and scale invariance of the log-polar mapping can be achieved after proper normalization [15]. Thus, we employ log-polar mapping to represent the non-uniform structure lying behind the retina in this paper.

Since the log-polar distribution is based on the polar coordinate system, the origin point and polar axis must be defined. However, how to determine the origin point and polar axis for a polar system is a major problem since the similarity between images can be calculated correctly only if all the similar images are translated to the same origin point and rotated to the same polar axis. The origin point could be estimated by the center of gravity of the
the entire foreground objects. The elongation axis of closed shape used in Ref. [19] was adopted as the polar axis in this paper. After that, the image can be transformed into the newly defined polar coordinate system. Meanwhile, the polar coordinates \((\rho(k), \theta(k))\) for each edge point can be derived. Let \(\mu\) and \(\sigma\) be the mean and standard derivation of \(\rho(k)\) values for all the edge points. Their values are important for determining the sampling rate of log-polar mapping as described later.

The log-polar mapping includes two steps. In the first step, the image is resampled into the log-polar grids denoted as the \(Z\) plane. In the second step, the \(Z\) plane is transformed into the Cartesian grids denoted as the \(W\) plane. Moreover, the mapping between the \(Z\) and \(W\) planes can be concisely defined by \(W = \log Z\) [16]. In other words, the \(W\) plane consists of a number of uniform grids, each of which represents a log-polar grid in the \(Z\) plane. An example of log-polar mapping is shown in Fig. 5. Note that circles about the origin of the \(Z\) plane map to lines parallel to the \(v(h)-axis\) in the \(W\) plane, i.e., horizontal line; radial lines in the \(Z\) plane map to lines parallel to the \(u(\log_2 \rho(k))-axis\), i.e., vertical line, where the base \(x\) of the log function will be defined below.

The parameters used for log-polar mapping are first described as follows.

1. \(N\): the numbers of samples per circle, set as 60 in this study;
2. \(E_0\): minimum sampled eccentricity;
3. \(L\): number of sampling circles;
4. \(E_i\): eccentricity of \(i\)th sampling circle, where \(i = 1, \ldots, L\) with \(E_i\) denoting the radial distance of the \(i\)th sampling circle from the center of gravity;
5. \(P_i\): width of the sampling grid in the \(i\)th sampling circle, where \(i = 1, \ldots, L\).

Two reasonable constraints were made in this paper. First, the sampling scope is between \(\mu - 2\sigma\) and \(\mu + 2\sigma\). Second, \(P_0\) cannot be less than one pixel. Since the perimeter of the \(i\)th sampling circle is \(NP_i = 2\pi E_i\), the width of the grid in the minimum sampling circle is \(P_0 = 2\pi E_0/N\). Thus, the minimum sampled eccentricity \(E_0\) modified from Ref. [16] can be calculated by the...
Fig. 5. An example of log-polar mapping. (a) The original image depicted on the polar coordinate system; (b) the corresponding $Z$ plane of (a); (c) the corresponding uniform $W$ plane of (a).

The eccentricity of the $i$th sampling circle is expressed by

$$E_i = E_0 \left(1 + \frac{2\pi}{N} \right)^i = E_{i-1} + P_{i-1}$$

where $i = 1, \ldots, L - 1$. (2)

Consequently, the base of the log$_x \rho(k)$ function can be derived as

$$\text{base} = x = 1 + \frac{2\pi}{N}$$
The width of the sampling grid in the $i$th sampling circle is
\[ P_i = \frac{2\pi E_i}{N} \quad \text{where } i = 0, 1, \ldots, L - 1. \]

Similarly, $E_L$ can be derived by the following equation:
\[ E_L = E_0 \left(1 + \frac{2\pi}{N}\right)^L < \mu + 2\sigma. \]  

Thus, we can obtain the following inequality:
\[ L < \log \left(\frac{\mu + 2\sigma}{E_0}\right) \log \left(1 + \frac{2\pi}{N}\right). \]

After the parameters $E_0$, $E_i$, $E_L$, and $L$ can be determined, respectively, by Eqs. (1)-(4) with fixed value $N$, all the log-polar grids in the $Z$ plane can be definitely depicted.

Consequently, only those pixels with $\rho(k)$ values between $E_0$ and $E_L$ in the original image will be sampled into a log-polar grid in the $Z$ plane which in turn become a Cartesian grid in the $W$ plane.

As shown in Fig. 5(b), the first circle with eccentricity $E_0$ has 60 samples in the form of $[0,1,1,1,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,\ldots,0]$, where 0 and 1 representing background color (white) and foreground color (blue), respectively. The sequence can be segmented into 4 horizontal stripes, namely $A_{1,1}$, $A_{2,1}$, $A_{3,1}$, $A_{4,1}$, and $A_{5,1}$, belonging to the first horizontal line in the $W$ plane as shown in Fig. 5. Actually, the pattern of consecutive subregions can be regarded as a horizontal profile. Several horizontal profiles of the same patterns can be further grouped into a horizontal stripe. In Fig. 5(c), each horizontal stripe is just a horizontal profile since no two consecutive horizontal profiles have the same pattern.

3. Retrieval based on 2D PHMM

The retrieval method consists of two major phases: construction of 2D PHMM and query comparison, which will be described in Section 3.1 and 3.2, respectively. Note that all the images, no matter the query or database images, must be resampled from their original images by log-polar mapping before the retrieval method is employed. In other words, we are concerned with the $W$ plane in this section.

3.1. Construction of 2D PHMM

The structure of a 2D PHMM includes two 1D HMMs associated with horizontal and vertical directions in the $W$ plane, respectively [12,13]. More specifically, each subregion of each stripe in the $W$ plane represents a state, while each stripe in the $W$ plane represents a horizontal 1D left-to-right HMM, which in turn becomes a superstate for a vertical 1D top-to-down HMM. A 2D PHMM constructed from the image in Fig. 5 is shown in Fig. 6. The relationships between Figs. 5 and 6 are illustrated through the subregion-to-state mapping.

The parameters of a 2D PHMM used in this study are listed here first.

\[ N^S = \text{the number of superstates in the model}; \]
\[ S^g = \{S_1^g, S_2^g, \ldots, S^g_n\} \text{ the set of superstates}; \]
\[ A^g = \{a_{j,1}, [1 \leq j, j_2 \leq N^g]\}, \text{ the superstate transition probability distribution}. \]

Each superstate $S^g_i \in S^g, 1 \leq j \leq N^g$ has the following parameters.

\[ N(j) = \text{the number of states in the superstate } S^g_i; \]
\[ S(j) = \{S_{1,j}, S_{2,j}, \ldots, S_{N(j),j}\}, \text{ the set of states}; \]
\[ v = \text{the observed value in each state}; \]
\[ A(j) = \{a_{l,j}(v) | 1 \leq i, i_2 \leq N(j)\}, \text{ the state-transition probability distribution}; \]
\[ B(j) = \{b_{l,j}(v) | 1 \leq i \leq N(j)\}, \text{ the observation probability distribution}. \]

In this study, the state-transition probabilities are heuristically defined on the basis of the width of the subregion [12,13] and can be expressed by

\[ a_{l,i}(j) = 1 - \frac{1}{\text{the width of the subregion } A_{l,j} + 1}, \]
\[ a_{l,i+1}(j) = (1 - a_{l,i}(j)) \times a_{l+1,i+1}(j), \]
\[ a_{l,i+2}(j) = (1 - a_{l,i}(j)) \times (1 - a_{l+1,i+1}(j)). \]

However, the superstate-transition probabilities are ignored since the number of superstates in the 2D PHMM for trademark is few. Moreover, the number of superstates allowed to jump can be a large number, say 4 in this study.

3.2. Query comparison

Actually, the most important part for a 2D PHMM approach is to obtain the observation probability distribution. Like Ref. [13], the observation probability for each state is interpreted as the color similarity between the grid in the $W$ plane of a query image and the subregion in the $W$ plane of a database image since each subregion represents a state of the corresponding 2D PHMM. Therefore, the color similarity between the grid $(x, y)$ in the query image and the state $S_{i,j}$ in the 2D
PHMM is newly defined as

\[
\begin{align*}
b_{i,j}(x,y) = \\
0.9 & \quad \text{if the grid } (x,y) \text{ and the state } S_{i,j} \text{ are both backgrounds}, \\
bf(i,j,x,y) & \quad \text{if the grid } (x,y) \text{ and the state } S_{i,j} \text{ are both foregrounds}, \\
0.0001 & \quad \text{otherwise}.
\end{align*}
\]

The function \(bf(i,j,x,y)\) is defined below and depicted in Fig. 7. Assume that the grid \((x,y)\) and the state \(S_{i,j}\) have the quantized colors \(C_{x,y}(Y_{x,y}, I_{x,y}, Q_{x,y})\) and \(C_{i,j}(Y_{i,j}, I_{i,j}, Q_{i,j})\), respectively. Let the Euclidean distance between \(C_{x,y}\) and \(C_{i,j}\) be \(D_{x,y}^{i,j}\). The function \(bf(i,j,x,y)\) can then be defined by the following equation:

\[
bf(i,j,x,y) = \begin{cases} 
0.05 (50 - D_{x,y}^{i,j})/50 + 0.95, & 0 \leq D_{x,y}^{i,j} \leq 50, \\
0.05 (100 - D_{x,y}^{i,j})/100 + 0.9, & 50 \leq D_{x,y}^{i,j} \leq 150, \\
0.4 (189 - D_{x,y}^{i,j})/189 + 0.5, & 150 \leq D_{x,y}^{i,j} \leq 340.
\end{cases}
\]

It is mentioned that the observation probability distribution will not be computed until the query image is given [13]. After the observation probability has been defined, the 2D Viterbi algorithm [10,12,13] can then be applied to estimate the probability value between a query image and a 2D PHMM in the database. The 2D version of the Viterbi algorithm [12,13] will be described as follows.

The 2D Viterbi algorithm for 2D PHMM

**Input:** A 2D PHMM \(\lambda\) with parameters defined in Section 3.1 and a query image \(Q\) of size \(X \times Y\).

**Output:** \(Pr[Q|\lambda]\), the log probability of the query image \(Q\) matched to the 2D PHMM \(\lambda\).

**Step 1:**
\[
\delta_{1,j}(1,y) = \log(b_{1,j}(1,y)), \quad 1 \leq j \leq N^s, \quad 1 \leq y \leq Y,
\]
\[
\delta_{1,j}(1,y) = -\infty, \quad 2 \leq i \leq N(j), \quad 1 \leq j \leq N^s, \quad 1 \leq y \leq Y,
\]
\[
\psi_{1,j}(1,y) = 0, \quad 1 \leq i \leq N(j), \quad 1 \leq j \leq N^s, \quad 1 \leq y \leq Y.
\]

**Step 2:**
\[
\delta_{x,j}(i,j) = \max_{i-2 \leq k \leq i} \left[ \delta_{x-1,j}(k,j) + \log(a_k(j)) \right]
\]
\[
+ \log(bf(i,j,x,y)), \quad 2 \leq x \leq X, \quad 1 \leq y \leq Y, \quad 1 \leq i \leq N(j), \quad 1 \leq j \leq N^s,
\]
\[
\psi_{x,j}(i,j) = \arg \max_{i-2 \leq k \leq i} \left[ \delta_{x-1,j}(k,j) + \log(a_k(j)) \right], \quad 2 \leq x \leq X, \quad 1 \leq y \leq Y, \quad 1 \leq i \leq N(j), \quad 1 \leq j \leq N^s,
\]
\[
1 \leq y \leq Y, \quad 1 \leq i \leq N(j), \quad 1 \leq j \leq N^s.
\]

**Step 3:**
\[
P_{y}(j) = \delta_{x,y}(N(j),j), \quad 1 \leq y \leq Y, \quad 1 \leq j \leq N^s.
\]
\[
\tilde{q}_{x,y}(j) = N(j), \quad 1 \leq y \leq Y, \quad 1 \leq j \leq N^s.
\]

**Step 4:**
\[
\tilde{q}_{x,y}(j) = \psi_{x+1,j}(\tilde{q}_{x+1,y}(j),j), \quad 1 \leq y \leq X - 1, \quad 1 \leq y \leq Y, \quad 1 \leq j \leq N^s.
\]

**Step 5:**
\[
P_{y}(j) = P_{y}(j) + \text{func}(y,j), \quad 1 \leq y \leq Y, \quad 1 \leq j \leq N^s.
\]

**Step 6:**
\[
\delta_{x}^{y}(1) = P_{1}(1),
\]
\[
\delta_{x}^{y}(j) = -\infty, \quad 2 \leq j \leq N^s,
\]
\[
\psi_{x}^{y}(j) = 0, \quad 2 \leq j \leq N^s.
\]

**Step 7:**
\[
\delta_{x}^{y}(j) = \max_{j-4 \leq k \leq j} \left[ \delta_{x-1}^{y}(k) \right] + p_s(j),
\]
\[
2 \leq y \leq Y, \quad 1 \leq j \leq N^s,
\]
\[
\psi_{x}^{y}(j) = \arg \max_{j-4 \leq k \leq j} \left[ \delta_{x-1}^{y}(k) \right],
\]
\[
2 \leq y \leq Y, \quad 1 \leq j \leq N^s.
\]

**Step 8:**
\[
P_{y}(N^s) = \max_{1 \leq k \leq N} \left[ \delta_{x}^{y}(k) \right],
\]
\[
\tilde{q}_{x}^{y} = \arg \max_{1 \leq k \leq N} \left[ \delta_{x}^{y}(k) \right].
\]

**Step 9:**
\[
\tilde{q}_{x}^{y} = \psi_{x+1}^{y}(\tilde{q}_{x+1}^{y}, y = Y - 1, Y - 2, ..., 1.
\]

**Step 10:** Stop.

The log probability function \(\delta_{x,y}(i,j)\) is computed for the horizontal profile \(y\) of \(Q\) and the superstate \(S_{j}^{y}\) of \(\lambda\). It follows, the log probability function \(P_{y}(j)\) can be computed to measure the proximity between the whole horizontal profile \(y\) of \(Q\) and the superstate \(S_{j}^{y}\) of \(\lambda\). Actually, \(P_{y}(j)\) will be used as the observation probability related to the superstate \(S_{j}^{y}\). Thus, the function \(\text{func}(y,j)\) is involved in Step 5 of the 2D Viterbi algorithm to incorporate the proximity between a horizontal profile in the query image and a superstate in a database image into matching score. Since a superstate of a 2D PHMM corresponds to a horizontal stripe in the database image, the proximity can be measured by horizontal characteristics. More specifically, two features of layout structure \(l(k)\) and foreground size \(f(k)\) based on the overall horizontal observation are derived by the
following equation [20]:

\[
l(k) = \sum_{i=0}^{N-1} m_k(i),
\]

\[
m_k(i) = \begin{cases} 
  i+1 & \text{if } p(i,k) \text{ is on the foreground object}, \\
  0 & \text{if } p(i,k) \text{ is on the background}, 
\end{cases}
\]

\[
f(k) = \sum_{i=0}^{N-1} S_k(i),
\]

\[
s_k(i) = \begin{cases} 
  1 & \text{if } p(i,k) \text{ is on the foreground object}, \\
  0 & \text{if } p(i,k) \text{ is on the background}, 
\end{cases}
\]

where \( p(i,k) \) is the \( i \)th pixel of the \( k \)th horizontal profile (stripe) in the \( W \) plane. The \( \text{func}(y,j) \) can then be defined as

\[
\text{func}(y,j) = \log_{10} \frac{\min(l(y),l(j)) + 1}{\max(l(y),l(j)) + 1} + \log_{10} \frac{\min(f(y),f(j)) + 1}{\max(f(y),f(j)) + 1},
\]
where \(l(y)\) and \(f(y)\) are the characteristics of the \(y\)th horizontal profile in the query image, while \(l(j)\) and \(f(j)\) are the characteristics of the \(j\)th horizontal stripe in the database image as defined by Eqs. (5) and (6). Examples to illustrate the layout structures and foreground sizes of horizontal profiles are shown in Fig. 8.

On the other hand, to record the optimal state sequence of the horizontal profile \(y\) matched to the super-state \(S^y\), an array of back states \(\psi_{x,y}(i,j)\) is used. The optimal sequence \(\hat{q}_1, \hat{q}_2, \cdots, \hat{q}_X\) can then be obtained from the state \(S_{N(i,j)}\) by backtracking \(\psi_{x,y}(i,j)\) form \(x = X - 1\) to 1. In the similar way, \(\delta_1(j), \delta_2(j), \cdots, \delta_Y\) can be defined and obtained using the 2D Viterbi algorithm [13]. Eventually, the similarity probability, \(P(Q|Z)\), between the query image \(Q\) and the 2D PHMM \(Z\) corresponding to a database image, can be directly computed, which is just \(P_{\text{hit}}(N^y)\).

It has to be mentioned that the function \(\text{func}(y, j)\) is not really tolerant in case of error in elongation axis orientation. In addition, as mentioned in Section 2.2, the elongation axis of foreground objects is chosen as the polar axis. Thus, if the deformation changes the elongation axis of the query image, it may lead to incorrect retrieval results.

To overcome the problems caused by the error of elongation axis, the 2D Viterbi algorithm must be performed many times at various starting columns to find the best matching score. However, the 2D Viterbi algorithm is computation intensive, thus for time reduction, only those columns of \(\theta\) values being multiples of 30° will be regarded as possible starting columns. In other words, the 2D Viterbi algorithm will be performed 12 times at starting columns being \(\theta = 0°, 30°, 60°, 90°, 120°, 150°, 180°, 210°, 240°, 270°, 300°\) and 330° provided that the \(W\) plane is wrapped around. It is noted that the radical line corresponding to the elongation (polar) axis in the \(Z\) plane is set as the first vertical line of \(\theta = 0°\) in the \(W\) plane, no matter query or database images.

Actually, the exhaustive search with 12 possible starting columns still needs expensive time. One efficient method may be directed to employ the vertical characteristics that are defined like horizontal characteristics. In other words, we can define layout structure \(l_i(k)\) and foreground size \(f_i(k)\) for each vertical line. The \(\text{func}_i(x, i)\) can then be similarly defined as

\[
\text{func}_i(x, i) = \log_{10} \frac{\min(l_i(x), l_i(i)) + 1}{\max(l_i(x), l_i(i)) + 1} + \log_{10} \frac{\min(f_i(x), f_i(i)) + 1}{\max(f_i(x), f_i(i)) + 1}
\]

where \(l_i(x)\) and \(f_i(x)\) are the characteristics of the \(x\)th vertical line in the query image, while \(l_i(i)\) and \(f_i(i)\) are the characteristics of the \(i\)th vertical line in the database image. Therefore, the starting column \(X\) of the query image can be defined as

\[
X = \arg \max_{-N_T \leq k \leq N_T} \left[ \sum_{w=1}^{N} \text{func}_i((k + w)\%N, (w)\%N) \right].
\]

(7)

where \(N_T\) is the tolerance of the elongation axis and \(N\) is the number of vertical lines. This solution is obviously efficient since exhaustive search has been restricted to the the tolerance range \(2 \times N_T\) and the matching

![Fig. 7. The depiction of the function hf(i, j, x, y).](image)

![Fig. 8. Examples to illustrate the layout structure and foreground size of horizontal profiles.](image)
algorithm is replaced by Eq. (7) rather than the 2D Viterbi algorithm.

On the other hand, since 2D Viterbi algorithm is computation intensive, we tried to reduce the computation load of the proposed method by using efficient features. First, in the quantization stage, the true colors of $256^3$ has been reduced to 36 color clusters. Second, in the log-polar sampling, the original image of sizes $128 \times 128$–$300 \times 300$ have been reduced to log-polar images of sizes $60 \times 5$–$60 \times 15$. It is noted that the sizes of trademark images are not fixed resulting in log-polar mapping images of variable sizes. Nevertheless, we can conclude that the spatial resolution can be reduced $(128 \times 128)/(60 \times 15) \approx 18–(300 \times 300)/(60 \times 5) = 300$ times. Accordingly, the computation load of 2D PHMM can be alleviated.

4. Experimental results

The proposed approach was implemented on a IBM compatible PC with a single Intel Pentium II-233 CPU and 192 Megabytes SDRAM, which is running Microsoft NT workstation Chinese version 4.0 operation system. The program was developed in C++ language and compiled under Microsoft Visual C++ version 4.2.

To prove the effectiveness of our method, we applied the proposed method on trademark database. The trademark database contained 210 monochrome and 191 color trademark images as listed in Ref. [21]. These trademarks are collected from books, VCD or homepage.

We chose 21 trademarks from the database to create five deformed trademarks for each of them. The 105 deformed trademarks were produced by manually redraw or the deformation options provided in the software of Paint Shop Pro version 4.10. The software package includes rotation, circle, cylinder, motion blur, pentagon, perspective, skew, wind, mosaic, etc. These deformed trademarks were regarded as the query images in the conducted experiment.

For each of the 105 deformed trademarks, five most similar trademarks were retrieved from the 401-image database. Retrieval accuracy was measured by the position (rank) of the required (original) trademark [6,9,22].
Ideally, the first retrieved trademark in the retrieval list should be the original trademark from which the query trademark is deformed. Thus, the ideal retrieval rank for each query must be 1. For the 105 deformed trademarks derived from 21 original trademarks, the respective retrieval ranks are listed in Table 2.

Since the origin of the log-polar mapping is properly chosen as the center of all the foreground objects as mentioned in Section 2.2, the achievement of translation invariance in our method is obvious thus not included here. Rotation and scale invariance of our approach is demonstrated in Figs. 9 and 10, respectively. Fig. 11 shows that our method can cope with composition resemblance. The overcome of deformation resemblance is illustrated in Figs. 12–14. On the other hand, some unsatisfactory results are shown in Figs. 15–17. The error sources may be from the two reasons listed below. First, the resampled results through log-polar mapping may be deformed since the origins of the original images and the resampling scopes may be varied. Second, although high tolerance of 2D PHMM can increase the flexibility of pattern matching, the order of retrieved trademarks may not match human perception of similarity.

5. Conclusions

A new deformed trademark retrieval method based on 2D PHMM is proposed in this paper. By our color quantization method, the number of colors are reduced from $256^3$ true colors to 36 colors, which in turn abates the requirements of storage and computing time. After that, the log-polar mapping takes into account the translation, rotation and scale variations. Thus, the proposed method...
Fig. 12. Sample query results. (a) Query by a deformed image; (b) a list of retrieved images ordered by similarity values.

Fig. 13. Sample query results. (a) Query by a deformed image; (b) a list of retrieved images ordered by similarity values.

Fig. 14. Sample query results. (a) Query by a deformed image; (b) a list of retrieved images ordered by similarity values.
Fig. 15. Unsatisfactory query results. (a) Query by a rotated image; (b) a list of retrieved images ordered by similarity values.

Fig. 16. Unsatisfactory query results. (a) Query by a magnified image; (b) a list of retrieved images ordered by similarity values.

Fig. 17. Unsatisfactory query results. (a) Query by a deformed image; (b) a list of retrieved images ordered by similarity values.
method can handle the invariant requirement. Moreover, the spatial resolution can be reduced 18–300 times. Obviously, the computation complexity of 2D PHMM can be alleviated. Finally, the flexibility for pattern matching can be increased since the 2D PHMM is high tolerance to noise as well as distortion and contextual information can be incorporated into the 2D PHMM in a natural and elegant way. However, 2D PHMM is computation intensive and sensitive to rotation, scale and translation variations. Thus, it is the main originality of this paper to include the advantages of 2D PHMM but to exclude its disadvantages. As a result, the composition and deformation resemblances can be coped with in the proposed method. Experimental results prove the effectiveness and practicability of our method. Further research may be directed to the following topics. First, parameter-learning strategy can be incorporated into the 2D PHMM to improve the retrieval performance. Second, the color quantization method can be generalized to be more adaptive. Third, the proposed method can be extended to natural images.

Acknowledgements

The authors would like to thank the anonymous reviewers for the valuable comments on the earlier version of this manuscript.

References


About the Author—MIN-TA CHANG graduated from National Taipei Institute of Technology, Taipei, Taiwan, ROC, in 1993 and received M.S. degree in Computer Engineering and Science from Yuan-Ze University, Taoyuan, Taiwan, ROC in 1999. Since 1999, he has been an Associate Researcher at Chunghwa Telecommunication Laboratories. His research interests include network management, image processing and pattern recognition.

About the Author—SHU-YUAN CHEN received the B.S. degree in Electrophysics in 1980, the M.S. and Ph.D. degrees both in Computer Engineering in 1982 and 1989, all from National Chiao Tung University, Hsinchu, Taiwan, ROC. From 1994, she has been an Associate Professor in the Department of Computer Engineering and Science of Yuan-Ze University, Taoyuan, Taiwan, ROC. Her research interests include image processing, pattern recognition, and image content retrieval.