Content-based image retrieval (CBIR) work includes selection, object representation, and matching. If a shape is used as feature, edge detection might be the first step of feature extraction. Invariance to translation, rotation, and scale is required by a good shape representation. Sustaining deformation contour matching is an important issue at the matching process.

In this paper, an efficient and robust shape-based image retrieval system is proposed. We use the Prompt edge detection method [19] to detect edge points, which is compared with the Sobel edge detection method. We also introduce a shape representation method, the low-to-high sequence (LHS), which is invariant to translation, rotation, and scale problems. The results of our proposed method show a superior matching ratio even in the presence of a modest level of deformation.

1. Introduction

Because of recent advances in computer technology and the revolution in the way information is processed, increasing interest has developed in automatic information retrieval from huge databases. In particular, content-based retrieval has received considerable attention and, consequently, improving the technology for content-based querying systems becomes more challenging. [11]

There are four important feature components for content-based image retrieval: color, texture, shape, and spatial relationship. Among these features, shape contains the most attractive visual information for human perception. An important step before shape extraction is edge point detection. Many edge detection methods have been proposed [8-10][15]. Z. Hou [8] uses a one-way design detector to determine existence of an edge point. J. Weickert [10] uses a partial differential and morphology method to formulate an optimization problem. M. Kass [18] has proposed an active contour model to surround the contour points.

They overcome the problem of multiple objects. Most of these research results show that they are robust but inefficient, because of the long processing time.

Another important issue next to edge detection is shape representation. A good shape representation method should be invariant to translation, rotation and scaling. F. Mahmoudi [11] uses an edge orientation autocorrelogram to represent an object to tolerate imperfect edge point detection. T. Bernier [2] records contour points of an object with their distances and angles relative to the center point of the object to form a shape representation, which is invariant to translation, rotation, and scaling, but yields a poor result when edge detection is incomplete. H. Nishida [3] proposed a representation that tolerates deformation of contour points. J. Zhang [7] uses a shape spaces method to deal with both the noise and occlusion problems. Methods in [3] and [7] both provide effective representations in practice, but the former is generally very expensive in terms of computational time, and the feature adopted by the latter is very sensitive to noise.

This paper is organized as follows: In Section 2, we bring up two commonly used edge detection methods and the one we shall use in our work. Section 3 proposes a new shape representation method that is invariant to translation, rotation, and scaling. This representation method is robust even when the contour is obviously deformed. In Section 4 the experimental results of the proposed method are compared with those of some other noted methods. The final section makes the conclusion for this paper.

2. Edge Detection

Edge detection is a necessary preprocessing step in most of computer vision and image understanding systems. The accuracy and reliability of edge detection is critical to the overall performance of these systems. Earlier researchers paid a lot of attention to edge detection, but up to now, edge detection is still highly challenging. In this section, we will briefly illustrate two common edge detection methods, and point out
their drawbacks. In addition, we introduce a simple and efficient method for edge detection.

2.1 Active Contour Model

The first common used method is the Active contour model, also called the Snake model [17]. The Snake model is an energy minimizing spline that can be operated under the influence of internal contour forces, image forces, and external constraint forces. An energy function of the Snake model can be defined using the following equation:[17]

\[ E = \int \left( E_{\text{internal}}[v(s)] + E_{\text{image}}[v(s)] + E_{\text{external}}[v(s)] \right) ds \]  

(1)

The location of the final template for the Snake model corresponds to a minimum of the energy function. However, the original Snake model meets some problems. The first problem is that without initially surrounding the object, the template will fail in detection. Second, the original Snake cannot detect multiple objects.

Wai-Pak Choi and Kin-Man Lam [8] proposed a modified version of the Snake model to solve these disadvantages. They used some critical points for splitting and connecting neighboring contour points. The idea is that if the distance between two adjacent contour points is greater than a given threshold, a new contour point is inserted between these two points. In contrast to the smaller distance between two adjacent contour points, a contour point between these two points is deleted. This method overcomes the problem of detecting multiple objects, but the need to initially surround the objects still remains, and it needs several iterations to meet the boundaries of objects and thus requires a lot of run time.

2.2 Sobel Operation

Another common edge detection method is the Sobel method [19]. The Sobel operation uses four masks to detect edge points, as illustrated in Figure 1.

The Sobel operation computes the partial derivatives of a local pixel to form the gradient. For the gray-level function \( f(x,y) \), the gradient of \( f \) at pixel \((x,y)\) is defined as a vector:

\[ \nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^T \]  

(2)

\[  0  1  1 \\
-1  0  1 \\
1  2  1 \]  

\[  0  1  1 \\
-1  0  1 \\
1  2  1 \]  

\[  0  1  1 \\
-1  0  1 \\
1  2  1 \]  

Figure 1. Sobel Masks

The magnitude of \( \nabla f \) can be approximated as the following equation:

\[ \| \nabla f \| = \left| (z_1 + 2z_2 + z_3) - (z_1 + 2z_2 + z_3) \right| \\
\| (z_1 + 2z_2 + z_3) - (z_1 + 2z_2 + z_3) \| \\
\| (z_1 + 2z_2 + z_3) - (z_1 + 2z_2 + z_3) \| \\
\| (z_1 + 2z_2 + z_3) - (z_1 + 2z_2 + z_3) \| \]  

(3)

Figures 1(a) to 1(d) show Sobel masks approximating the derivatives in the x direction, the y direction, and the diagonal directions, respectively. Equation (3) computes the magnitude level of the current interesting pixel, if \( \| \nabla f \| \) is greater than a threshold, this pixel is marked as an edge point.

Although Sobel’s edge detector generally works well, it must repeatedly go over the whole image many times. As a result, this method requires \( W \times H \times 3 \) time to operate on an image of size \( W \times H \). We shall compare this result with that of the proposed method in Section 2.3.

2.3 A Prompt Edge Detection Method

In this subsection, an easy and effective edge detection method, the Prompt edge detection method, is introduced.

The Prompt edge detector detects edge points by checking the differences in the gray value for each point from those of its neighbors. For a pixel \((x,y)\) with gray value \( g(x,y) \), let \( g_0(x,y), g_1(x,y), \ldots, g_7(x,y) \) denote the gray values of its neighbors in 8 directions as shown in Figure 2.

\[ g_3 g_2 g_1 \]
\[ g_4 g g_0 \]
\[ g_5 g_6 g_7 \]

Figure 2

Let \( h_d(x,y) = \left| g(x,y) - g_d(x,y) \right| \) be the difference in the gray value of pixel \((x,y)\) from that of its neighbor in direction \( d \). Let \( B_d(x,y) \) be the number of differences that exceed a threshold \( T \), where \( T = a + c \), \( c \) is a constant, and \( a \) is the average difference between all pairs of adjacent pixels’ gray values in the image. In this work, we take the value 2 for \( c \) and, instead of taking a single value for \( T \), we take the local average differences for \( T \) by dividing the whole image into
blocks and computing the average difference for each block.

The pixel \((x, y)\) is indicated as an edge point if the following inequalities hold:

\[
3 \leq B_3(x, y) \leq 6 \tag{4}
\]

These inequalities can also avoid indicating noisy points as edge points.

3. Shape Representation

A good representation can manifest an object’s characteristics. Furthermore, it would help achieve a high recognition rate for a content-based image retrieval system. Some researchers [1-4] have shown that objects are very distinguishable based on their visible features. Among these features, shape is the most important for recognizing objects. In this section, we introduce a new shape representation method that is invariant to translation, rotation, and scaling, and also has high tolerance to complex, occluding, or even deformed images.

3.1 Central Point Determination

The first step in shape representation for an object is to locate the central point of the object. To permit invariance to translation, rotation and scaling, the geometric center of the object is selected as a reference point.

We use the equation (5) to compute the centroid of an object.

\[
x_c = \frac{\sum_{i=1}^{n} x_i}{n}
\]

\[
y_c = \frac{\sum_{i=1}^{n} y_i}{n}
\]

where \(n\) is the number of points of an object.

3.2 Polar Representation and Distance Sequences

A shape representation method [3] is outlined in this section. We characterize the contour using a sequence of contour points described in polar form. Here, we take the pole at the centroid \((x_c, y_c)\), then the contour graph can be represented by a polar equation \(d = f(\theta)\), and each contour point \((x, y)\) has the polar description \((d, \theta)\), where \(x, y, d,\) and \(\theta\) are related using Equations (6) and (7). A sketch map is shown in Figure 3.

\[
d = \sqrt{(x-x_c)^2 + (y-y_c)^2} \tag{6}
\]

\[
\theta = \tan^{-1}\left(\frac{y-y_c}{x-x_c}\right) \tag{7}
\]

With the polar description, we may represent contour points as a sequence \((d_0, d_1, \ldots, d_{n-1})\), where \(d_i = f(\theta_i)\) and \(\theta_i = i \times \frac{2\pi}{n}\). Using Equation (6), \(d_i\) is known to be the distance between the corresponding contour point and the centroid. Figure 4 illustrates the graph of the polar equation \(d = f(\theta)\) for a given contour.

We can obtain the distance sequence \((d_0, d_1, \ldots, d_{n-1})\) by successively rotating a ray emanating from the pole counterclockwise through a fixed angle \(\Delta \theta\), where \(\Delta \theta = \frac{2\pi}{n}\) for a positive integer \(n\), and at each step of rotation, recording the distance of intersection point of the current ray \(L_i\) with the contour. This method of representing and describing the shape of an object is adequate for convex objects. However, for an object containing concavities, a ray may intersect the contour with more than one point. If we only record the distance of one point (say, smallest distance) at each step of rotation, the contour points of some protrusions may be missed, see Figure 5. We can eliminate this problem by recording the distance of the farthest intersection point at each rotating the scanning ray step, or for a more detailed description, recording the distances, or furthermore associating the number of all the intersection points. With this idea, as the result shown in Figure 6, the farthest intersection points C and D, instead of A and B as shown in Figure 5(a), found during the horizontal ray scanning step, as thus this object can be distinguished from the one in Figure 5(b). To provide scale invariance, the maximum distance is
computed and all distances are normalized to it. Thus, all values fall between 0 and 1 regardless of how much the objects are scaled.

3.3 Mountain Climbing Sequence (MCS)

Since we used the pole at the centroid of the object, the representation of a distance sequence is translation-invariant. To achieve rotation-invariance, Bernier [2] rotated the polar coordinate system, such that the maximum distance was associated with the angle of zero. However there is no any guarantee for this method if there are more than one maximum distance. The representation will not be unique. In the example shown in Figure 7, both objects have three maximal distances and each has three possible representations: $(\sqrt{2} , 1, 1, \sqrt{2} , 1, \sqrt{2} , 1, 1)$, $(\sqrt{2} , 1, \sqrt{2} , 1, \sqrt{2} , 1, 1)$, and $(\sqrt{2} , 1, \sqrt{2} , 1, 1, \sqrt{2} , 1, 1)$. To reduce this problem, we propose another representation method that deals with all distances rather than only the individual maximal distances in the sequence. First, we evaluate the ordering-consistency functions $c_i$'s, as defined in Equation (8), at the sequence $D = (d_0, d_1, \ldots, d_{n-1})$ as described in sub-section 3.1, and determine the index, $s$, of the function having the smallest value, as defined in Equation (9). The distances in the sequence $D$ are then shifted forward for $s$ positions to yield a new sequence $D_m=(d_s, d_{s+1}, \ldots, d_{s+n-1})$, where $d_k = d_k \mod n$ for each $k$. This new sequence $D_m$ is called the mountain climbing sequence (MCS) for the object.

The MCS representations for Figures 7(a) and 7(b) are identical and both come out as $(1, 1, \sqrt{2} , 1, \sqrt{2} , 1, \sqrt{2} )$, and as a result, they are considered having the same shape.

3.4. Shape Matching

The final step of content-based image retrieval is the matching stage. As discussed in the preceding sub-section, we use the MCS to represent the shape of objects. We proceed to the matching stage to measure the similarity between two objects by simply evaluating the Euclidean distance between each object’s MCSs of them. The Euclidean distance between two sequences is given in Equation (10).

$$d(U, V) = \sqrt{\sum_{i=0}^{n-1} (v_i - u_i)^2}$$  \hspace{1cm} (10)

where $U$ and $V$ are the query and database images, $v_i$ and $u_i$ are their $i$th features, respectively, and $n$ is the dimension of the feature space. Any arbitrary two-dimensional shape can then be compared to any other using the outline method.

4. Experiments and Results

This section consists of two subsections. The first subsection compares the prompt edge detection method with Sobel operation. The second subsection presents our contour retrieval results.

4.1 Edge Detection Results

In this subsection we demonstrate the results of our edge detection method. We also compare the processing time with that of the Sobel operation. Figure 10 shows some results from our proposed method. The test images include binary, gray, and true color images.

For a color image, we transfer the RGB color model into the HSI color model and use the I vector to perform edge detection. Figures 8 (c) and (d) show such an example. Figure 9 shows the run time for the Sobel and our proposed method. We find that with growing image size, the time efficiency of the proposed method becomes more evident compared with that for the Sobel’s method.
4.2 Contour Retrieval Results

The shape feature database was constructed for the image data set publicly available through the web site: http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html from the VSSP Center of the University of Surrey, UK. The data set is composed of 1100 marine creature image boundary contours, originally scanned from some printed books. Some experimental shape retrieval results are presented in Figure 10. In each figure, the query image is presented at the top, and the query results are arranged from the upper left to the lower right corner in ascending distance order.

Figure 10. Example of shape retrieval from the shape database of marine creature images. The query shape is presented at the top, and retrieved results are arranged from the upper left to the lower right corner in ascending distance order.

The results compared with those of H. Nishida [3] are shown in Table 1. Table 1 lists the average query rates in top 1%, 2%, 3%, 4%, 5%, and top 10% of outcome, respectively, with query images in different degrees of rotation and different deformation level. The deformation rate means the percentage of deformation in the contour of an object. We can see from the table that the results from the proposed method is better than that by Stein’s method, and that the proposed method has much higher tolerance with noise than Nishida’s method.

5. Conclusions

This paper presented a new edge detection method [18] that reduces the processing time compared with the Sobel method. We also proposed a new shape representation method, the mountain climbing sequence, to meet the invariance to translation, rotation, and scaling requirement. In addition, the experimental results show that the proposed method highly tolerates complex, deformation and occluding images. We also compared our proposed method with Nishida and Stein’s methods [3] to prove its robustness and effectiveness.

6. References

International Conference on Computer Graphics and Imaging, August, 2001 13-16, Honolulu, Hawaii, USA.


Table 1
Average query rates (%) of deformed patterns of Proposed, Nishida and Stein methods.

<table>
<thead>
<tr>
<th>Rotation</th>
<th>Deformation (%)</th>
<th>Method</th>
<th>r=1%</th>
<th>r=2%</th>
<th>r=3%</th>
<th>r=4%</th>
<th>r=5%</th>
<th>r=10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°</td>
<td>0</td>
<td>Proposed</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nishida</td>
<td>92.1</td>
<td>96.1</td>
<td>97.6</td>
<td>98.2</td>
<td>98.7</td>
<td>99.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stein</td>
<td>92.8</td>
<td>96.8</td>
<td>98.3</td>
<td>98.6</td>
<td>98.8</td>
<td>99.7</td>
</tr>
<tr>
<td>10°</td>
<td>20</td>
<td>Proposed</td>
<td>98.1</td>
<td>99.2</td>
<td>99.9</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nishida</td>
<td>96.6</td>
<td>99.7</td>
<td>99.7</td>
<td>99.7</td>
<td>99.8</td>
<td>99.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stein</td>
<td>86.3</td>
<td>92.3</td>
<td>94.8</td>
<td>96.1</td>
<td>97.3</td>
<td>98.9</td>
</tr>
<tr>
<td>90°</td>
<td>0</td>
<td>Proposed</td>
<td>96.3</td>
<td>98.7</td>
<td>99.9</td>
<td>99.9</td>
<td>99.9</td>
<td>99.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nishida</td>
<td>99.8</td>
<td>99.8</td>
<td>99.8</td>
<td>99.9</td>
<td>99.9</td>
<td>99.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stein</td>
<td>80.6</td>
<td>87.4</td>
<td>89.9</td>
<td>92.1</td>
<td>92.8</td>
<td>96.5</td>
</tr>
<tr>
<td>10°</td>
<td>20</td>
<td>Proposed</td>
<td>97.2</td>
<td>98.6</td>
<td>99.9</td>
<td>99.9</td>
<td>99.9</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nishida</td>
<td>99.2</td>
<td>99.6</td>
<td>99.7</td>
<td>99.7</td>
<td>99.7</td>
<td>99.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stein</td>
<td>76.8</td>
<td>84.6</td>
<td>88.7</td>
<td>90.5</td>
<td>91.6</td>
<td>95.8</td>
</tr>
<tr>
<td>20°</td>
<td>20</td>
<td>Proposed</td>
<td>96.1</td>
<td>98.2</td>
<td>99.1</td>
<td>99.5</td>
<td>99.9</td>
<td>99.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nishida</td>
<td>96</td>
<td>97.8</td>
<td>98.2</td>
<td>98.6</td>
<td>99.1</td>
<td>99.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stein</td>
<td>69.5</td>
<td>79.8</td>
<td>83.9</td>
<td>86.4</td>
<td>88.1</td>
<td>93.1</td>
</tr>
</tbody>
</table>