Shape Representation of Clipart Image with Concave and Convex

Chang-Gyu Choi*; Yongseok Chang*; Sang-Ryul Ryu**; and Sung-Ho Kim*

* Multimedia Laboratory, Department of Computer Engineering, Kyungpook National University, Daegu, Korea
** Department of Computer Science, Chungwoon University, Chungnam, Korea

Abstract

This paper presents a method of extracting shape information from clipart images and then measuring the similarity between different clipart images using the extracted shape information. Previous methods of extracting shape information can be classified into outline-based methods and region-based methods. Included in the former category, the proposed method expresses the convex and concave aspects of an outline using the ratio of a rectangle. Experimental results demonstrated that the proposed method was superior in expressing shape information than previous outline-based feature methods.

1. Submission of Manuscripts

With the increasing use of the Internet and the rapid development of multimedia technologies, there is also a need for technologies that can automatically extract, store, transmit, and search for multimedia data, such as images, moving pictures, or audio data. As such, image content-based retrieval systems have already been developed that can automatically extract, store, and search for images using color, texture, and shape [1,2,3,4,5]. The extraction methods that use shape features can basically be categorized as either outline-based methods or region-based methods [1, 3, 4].

Among As regards outline-based methods: Freeman and Davis were the first to represent the outline of an arbitrary shape using a Chain Code String in a clockwise or counterclockwise direction [1], [6]. However, since a Chain Code String will vary according to rotation and size, several modified methods have also been introduced, including the Derivative Chain Code proposed by Bribiesca and Guzman [1], [7]. Subsequently, Jain propos-ed a Fourier transform method where points on a 2-D plane, which are selected to outline a shape, are transform-ed into coefficient vectors using a Fourier transform [1], [8]. Finally, Rauber and Steiger-Garcao proposed the UNL Fourier Features method [1], [4], whereby a UNL Fourier transform is used to change the Descartes coordinates of a shape into polar coordinates. The UNL Fourier features are invariant to size and translation, yet variant to rotation. A 2-D Fourier transform is thus used to solve this problem. Accordingly, to deal with the weaknesses in the previous methods, this paper proposes a new method that expresses

the convex and concave aspects of an outline using the ratio of a rectangle, thereby expressing superior shape information than previous outline-based feature methods.

The remainder of this paper is organized as follows. Section 2 outlines the proposed system and preprocessing methods. Section 3 describes the shape extraction method, rectangle representation, chain code, color representation and estimation of similarity. Section 4 presents the experiment results, and section 5 gives some final conclusions and suggests areas for future work.

2. Proposed System and Preprocessing Steps

2.1 Outline of Proposed System

The proposed system extracts shape information from a clipart image, saves a database (offline process), and then measures the similarity between clipart images (online process). Fig. 1 shows the structure of extraction to shape information.

![Fig. 1 Diagram of extraction to shape information](image-url)

The offline process consists of two preprocessing steps: extraction of clipart image outline and its conversion into a polygon. The proposed rectangle representation is represented using the polygon information, while the information is changed into a chain code. The online process then measures the similarity between the features of the query clipart image, extracted in the same way as the offline process, and those in a database.

2.2 Extraction of Outline and Conversion into Polygon
Fig. 2 shows the original clipart image and a polygon image, which is the extracted outline shape. To extract the outline, a start point pixel is selected in the left-top part of the original image, then sampling pixels are extracted in proportion to the ratio of the image size, to make a polygon. The sampling interval is determined by equation (1).

\[
S = \frac{W + H}{20}
\]

where, \( S \) is the sampling interval, \( W \) is the image width, and \( H \) is the image height.

Fig. 3 shows a flow diagram of the outline extraction and creation of a polygon. When a candidate pixel is identified in the sampling interval, this pixel is included in the polygon vertexes. The procedure stops when the candidate pixel is the same as the starting pixel. Unlike human eyes, an image outline includes noise, which is removed through the sampling interval. Yet, there is a trade-off, as if the sampling interval is too small, the noise will be not be removed.

3. Extraction of Shape Information and Measurement of Similarity of Manuscripts

3.1 Rectangle Representation

After extracting the outline and converting the original image into a polygon image, as mention in section 2.2, the proposed method uses a rectangle representation to express the convex or concave sections of the polygon image. Fig. 4 (a) and (c) show part of a polygon image, while (b) and (d) show the respective rectangle representations of the concave regions. In Fig. 4 (a) and (c), the concave regions are similar, but (a) is more concave than (b). This difference between the concave regions is clearly represented when using the rectangle representation, even though the shapes are both concave regions. The same method is also adopted for convex regions.

Fig. 5 shows a flow chart of how the rectangles are created from the shape information. Before initiating the steps in Fig. 5, it is determined whether the polygon’s vertexes are convex or concave in a clockwise direction. The vertex is convex when the next vertex is located to the right in a clockwise direction, otherwise the vertex is concave, i.e., the next vertex is located to the left.

3.2 Construction of Chain Code

To measure the similarity between the shapes of clipart images, the current study uses a chain code. The chain code is constructed from the rectangle representation using two symbols, the first representing a convex rectangle and the other representing a concave rectangle, plus a real value that represents the ratio of the width to the height of the rectangle. The real value is used to express the extent of the convexity or concavity, as shown in Fig. 4 (b) and (d).

C is used to represent a concave rectangle, while \( V \) represents a convex rectangle. The real value is calculated
based on the ratio of the height of the rectangle, \( H \), to the width of the rectangle, \( W \), i.e. \( H/W \). The width and height are calculated as presented in Fig. 5.

Fig. 6 shows an example of a rectangle representation and the associated chain code. The chain code is constructed from the start rectangle following a clockwise direction and consists of the real value \( a_i \), which is the ratio value, and the symbol \( C \) or \( V \) related to each rectangle. As such, the chain code is constructed based on an arrangement of real values and symbols.

\[
a_1 C a_2 V a_3 C a_4 V \cdots a_{n-1} C a_n V
\]

\( n \): integer, \( a_i(1 \leq i \leq n) \): ratio of rectangle, i.e. \( H/W \), and \( C \), \( V \): symbol representing convex or concave rectangle, respectively.

\[V 0.4 C 0.8 V 0.1 C 0.6 V 0.3 C\]

\( n = 6 \), \( i = \{1, 2, 3, 4, 5, 6\} \)

\[W_c = \frac{1}{N} \sum_{i=0}^{3} \sqrt{(H_i - H'_i)^2 + (S_i - S'_i)^2 + (V_i - V'_i)^2}\]

where, \( CS = \{H_0, S_0, V_0, \ldots, H_3, S_3, V_3\} \) and \( CS' = \{H'_0, S'_0, V'_0, \ldots, H'_3, S'_3, V'_3\} \)

### 3.3 Color Representation

For extraction of representative colors on a clipart image, we adopt a HSV color retrieval system. To calculate the distance between colors, we quantize a set \((H, S, V)\) to \((H', S', V')\) and we select the largest 4 color sets in clipart image. The equation (2) is illustrated the quantization function.

\[H' = \text{int}(H/11.25) \times 11.25\]

\[S' = \begin{cases} 0.5, & \text{if } S \geq 0.5 \\ 0, & \text{otherwise} \end{cases}\]

\[V' = \begin{cases} 0.2, & \text{if } V \geq 0.2 \\ 0, & \text{otherwise} \end{cases}\]

The elements in \( CS \) and \( CS' \) are sorted by descending and \( N \) is 4.

\[W_c = \frac{1}{N} \sum_{i=0}^{3} \sqrt{(H_i - H'_i)^2 + (S_i - S'_i)^2 + (V_i - V'_i)^2}\]

### 3.4 Measurement of Similarity

Using the vectors in the chain code, the similarity step is as follows. First, the same chain code length is created for comparative clipart images, since the length of two clipart images may be different or the same. Second, the similarity between a query clipart image and the database is measured. To make the same chain code length for two images, the length of the longer chain code between two clipart images is shortened, thereby modifying its shape. In the current paper, to make the same length, concave regions are eliminated, as shown in Fig. 7.

\[V C a V a \cdots V a\]

\( k \): integer, \( a(1 \leq a \leq k) \): ratio of rectangle, i.e. \( H/W \), and \( C \), \( V \): symbol representing convex or concave rectangle, respectively.

\[\text{The weight must be the minimum value to change equation (4) to equation (5). To obtain the minimum weight, } d \text{ is}\]

\[d \text{ is the minimum value to change equation (4) to equation (5). To obtain the minimum weight, } d \text{ is}\]

\[d \text{ is the minimum value to change equation (4) to equation (5). To obtain the minimum weight, } d \text{ is}\]
determined as the minimum sum of the distance between \( d \) and each point, \( a_{k,i}, a_k, \) and \( a_{k+1} \). The minimum weight is thus the minimum cost to modify a concave region, which is located in the center, and the convex regions, which are located at the end-to-end sides. Plus, \( a_{k,i}, a_k, \) and \( a_{k+1} \) are considered as a vector, so the distance is calculated between two vectors \([1, 4, 9]\). The rotation of the chain code solves the rotation variation problem with a chain code, and the use of the two characters is to match the same region, i.e. a convex region or concave region.

Let one chain code be \( S_i \), the other chain code that is rotated to \( i-th \) be \( S_2 \), and the length of the chain code be \( n \).

The similarity is then defined using equation (7), where \( W_s \) is a value between 0 and 1.

\[
S_1 = a_1C a_2V \cdots a_{n-1} C a_n V, \quad S_2 = b_1C b_2V \cdots b_{n-1}C b_n V
\]

\[
v_1 = (a_1, a_2, \cdots, a_n), \quad v_2 = (b_1, b_2, \cdots, b_n)
\]

\[
d_i = \frac{1}{1 + \sqrt{\sum_{j=1}^{n} (a_j - b_j)^2}}
\]

\[
W_s = \min\{d_i\}, \quad i = 1, 2, \cdots, n
\]

The final similarity between a query clipart image and the database is the sum of \( W_t \) and \( W_r \) and \( W_c \), thus equation (8) is the similarity.

\[
W = w_1 \times \left( \frac{W_r + W_c}{2} \right) + w_2 \times W_r, \quad \text{where} \, w_1 + w_2 = 1
\]

4. Experimental Results

The experiment involved 500 clipart images, consisting of 100 original images, along with the same 100 images reduced 25% and 50%, and rotated 90° and 315°. The weight \( w_1 \) and \( w_2 \) is 0.7 and 0.3 each other. To compare with other methods, a Chain Code String and Fourier Descriptor were selected and the performance results summarized in terms of the Precision and Recall [1].

The query image was selected from the 100 original images and the performance was calculated based on the average Precision and Recall value. The 5 images with the closest similarity were selected. Fig. 10(a) and (b) present the experimental results and Table 1 presents the results compared with the other methods. Table 1 shows that the proposed method was superior to the other outline-based methods and more insensitive to a variation in the outline.

The Precision and Recall for the proposed method was lower for images 6 and 8 than with the other methods due to the preprocessing method. Since variations in the outline...
are disregarded, the proposed method can be more efficient when modifying the preprocessing steps.

**Table 1.** Experimental results compared with other methods (selected 10 images)

<table>
<thead>
<tr>
<th></th>
<th>Proposed method</th>
<th>Chain code string</th>
<th>Fourier Descriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Image 1</td>
<td>1.00</td>
<td>1.00</td>
<td>0.60</td>
</tr>
<tr>
<td>Image 2</td>
<td>0.80</td>
<td>0.80</td>
<td>0.60</td>
</tr>
<tr>
<td>Image 3</td>
<td>1.00</td>
<td>1.00</td>
<td>0.60</td>
</tr>
<tr>
<td>Image 4</td>
<td>0.60</td>
<td>0.60</td>
<td>0.80</td>
</tr>
<tr>
<td>Image 5</td>
<td>0.60</td>
<td>0.60</td>
<td>0.80</td>
</tr>
<tr>
<td>Image 6</td>
<td>0.40</td>
<td>0.40</td>
<td>0.40</td>
</tr>
<tr>
<td>Image 7</td>
<td>0.80</td>
<td>0.80</td>
<td>0.40</td>
</tr>
<tr>
<td>Image 8</td>
<td>0.40</td>
<td>0.40</td>
<td>0.40</td>
</tr>
<tr>
<td>Image 9</td>
<td>0.80</td>
<td>0.80</td>
<td>0.20</td>
</tr>
<tr>
<td>Image 10</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
</tr>
<tr>
<td>Average(10)</td>
<td>0.70</td>
<td>0.70</td>
<td>0.54</td>
</tr>
<tr>
<td>Total (500)</td>
<td>0.63</td>
<td>0.63</td>
<td>0.47</td>
</tr>
</tbody>
</table>

(P : Precision / R : Recall)

![Query image](image)

**Fig. 10** Experimental result

5. Conclusion

The current paper designed and implemented a system that can extract shape information from a clipart image, save the information in a database (offline process), and measure the similarity between clipart images (online process). The proposed system consists of three steps, creating a polygon image from the original image, rectangle representation of the polygon image, and changing the chain code. The rectangle representation provides a more detailed representation of the shape information than other outline-methods. However, it is difficult to apply the proposed system to natural images, due to problems extracting the outlines, therefore, the new system is presently limited to clipart images. Accordingly, further studies will focus on other features, such as color, texture, etc.

**References**


