A Novel Method for Object Shape Description with Discrete Segments

MAHABOOB SHARIEF SHAIK (*)
G.SHYAMA CHANDRA PRASAD (**)
shyam_gunda2002@yahoo.co.in

ABSTRACT

Image representation is one of the most important topics of digital image processing. Two dimensional object shape description has gained great attention in content-based image retrieval (CBIR) and in many applications that handles with the shapes of objects. Object shape descriptors are classified into two forms: contour-based and region-based. In this paper we present contour-based shape representation and description, where the object shape is represented and describe as a set of discrete segments. The descriptor is invariant to translation, scale and Circular shifting makes it robust to rotation.

1. Introduction

Multimedia dominates the lifes of people. Due to The expansion of multimedia information, a challenging problem of designing techniques that support effective searching through large volume image databases has risen. In response to such a challenge, much research has been done to develop tools for easy and effective access to image databases. Image database querying has become a major area of research and received increased attention.

Image retrieval systems rely on Text-based and /or content-based methods in order to determine which images must be retrieved from the database. Techniques using textual attributes for annotations are common to text retrieval applications. In the past few years, a large number of commercial image data management systems have been developed. Most of these commercial systems are only capable of retrieving images based on pure textual description and this despite the fact that images have attributes which set them apart from text and which make provision for access far from simple.

(*) "Faculty of Computing & Information Technology King Abdul Aziz University, Kingdom of Saudi Arabia
(**) Professor,Ace Engineering College,India
Eakins defined Content-based image retrieval CBIR as a "technique for retrieving images on the basis of automatically-derived features such as color, texture and shape". He also argued that the extraction process must be predominantly automatic and the retrieval of images by manually assigned keywords is not CBIR even if the keywords describe image content. CBIR is not a replacement but a complementary component to text-based image retrieval; a satisfactory retrieval performance requires the integration of both of them. Because shapes bear semantic meanings, Shape is the most important feature among all features that are used for querying the content of images. However, shape representation and description is also considered as one of the most difficult aspects of content-based retrieval, this is because shape is often damaged with noise, many systems are relatively successful in retrieving by color and texture, but perform poorly when searching on shape.

Commonly shape representation and description looks for effective ways to capture the shape features that make it easier for a shape to be stored, transmitted, compared against and recognized. These features must also be independent of translation, rotation and scaling of the shape. In many cases human vision system can recognize objects based on their shapes, these shapes can be obtained by tracking their boundaries. The issue here is how to represent a shape using its extracted boundary. A survey in shows that users are more interested in retrieval by shape than by any other features such as color and texture. Many shape representation and description techniques have been developed in the past.

![Classification of Boundary based shape representation](image)

There are many shape representation and description techniques available. Zhang
et al. in classified Shape representations and descriptions methods into two
categories: boundary-based (also called Contour-based) and region-based, these
methods are performed by using defined image-coding schemes to represent image
contents. In boundary-based, objects are represented in terms of their external
characteristics, i.e. the pixels along the object boundary, while in region-based:
shape features are extracted from the whole shape region, i.e. the pixels contained
in the region. The different methods under boundary-based category are further
divided into two sub-categories containing global and structural methods.

The most successful representatives for global boundary-based method is Fourier
Description, Fourier Descriptor is obtained by applying Fourier transform on
shape boundary. It was proposed by Persoon and Fu in 1977. Many research have
been done based on Fourier Descriptor. However, Fourier Descriptor is sensitive
to the starting point of the shape boundary, if the starting point changes the whole
boundary sequence will change. Since our work falls under Structural boundary-
based method, in the following section we focus our attention on this type of shape
representation and description.

**Structural boundary-based method:** With Structural– based shape
representation and description, shapes are broken down into boundary segments
called primitives. Different Structural methods use different ways in selecting and
organizing the primitives in a form suitable for further computer processing. For
example: the primitives can be extracted based on common methods such as,
polygonal approximation, curvature decomposition and curve fitting, then encode
and organize the extracted primitives in a form of strings or graphs that can be
directly used for description or as input to a higher-level syntactic analyzer. The
result of this type of representation is encoded into a string of the general form:

\[ P = p_1, p_2, \ldots, p_n \]  

Where \( p_i \) may be an element of a chain code, a side of a polygon, a spline, etc. \( p_i \)
may contain a number of attributed like length, average curvature, maximal
curvature, orientation etc.

Many techniques exist utilizing Structural–based shape representation, such as
boundary approximations, chain codes, scale-space techniques and syntactic
techniques. Chain code is one of the early boundary representation attempts. It
was introduced in 1961 by Freeman. Chain code describes an object by a
sequence of directional vector (unit-size line segments with orientation). Chain
code suffers from digitization noise so it is not desirable to use it directly for
shape description and matching. Furthermore Chain code is not rotation
invariant by nature, to overcome this problem the first difference of the chain
code is used instead of the code itself.

Boundary approximations can be achieved by polygonal approximations.
Polygonal approximations are used to approximate the shape boundary using
polygonal line such that a global approximation error is minimized. Split and
merge is one of the well known methods in this group, Split-and-merge method.
splits the boundary into segments iteratively until each curve segment can be approximated by linear segments within a predetermined error range and merge the adjacent segments to form a single segment if a predetermined error is not exceeded, this method can approximate any shape with good accuracy compactly if the object has straight edges. The difficulty with this method is that vertices in the resulting polygon don’t always correspond to inflections points in the original boundary.

In scale-space approach, the shape is represented by the position of inflection points of the boundary, which remain present after passing several Gaussian filters of various widths over the shape boundary. These Stable inflection points are expected to be the main shape characteristics. Syntactic shape analysis is based on the theory of formal language. In syntactic methods, shape is represented symbolically by a set of predefined primitives, the set called the codebook and the primitives are called code words. It aims to create a language that can describe shapes by sentences these sentences are strings of symbols. This method needs a priori knowledge of the shapes in the database in order to define the list of code words.

**Method overview:** Our own research efforts are directed towards searching image databases for shapes, which are similar or partially similar to the query shape. It is worth mentioning that the proposed approach is mainly designed to query for man-made objects (e.g. cars, buildings, tools, etc.)

Since query shape may not exactly match any of the shapes in database, using approximate representation and description should be sufficient. Shape in database with minimum difference from the query shape is considered the best match to the query. In our solution, we only exploit shape boundary (outline), which will be decomposed into discrete segments in order to reduce the boundary complexity and simplify the process of information extraction. Next we will introduce our method for 2D object shape representation and description.

**Shape representation:** The most straightforward way to describe a shape is by using information from its boundary. Digital boundaries tend to be distorted and complicated (ragged) due to digitization noise and filtering errors, so obtaining reliable information from such boundaries is considered as a very difficult process.

The soul of this method is to neglect the distortions by decomposing the 2D object boundary into sequence of straight-line segments (lengths and directions), which lead to generate an approximate representation of the original boundary of the object. This representation must preserve the perceptual appearance of the original boundary shape at a sufficient level while eliminating the boundary distortions. In other word, our goal here is to capture the essence of the original boundary shape with the fewest possible segments. For simplicity, we assume extracting only the boundary outline of an object with single-pixel thick (8 connected curves), generating what we call boundary-image, this step is fully automated with today’s technology. The boundary-image is divided into fixed size blocks (e.g. 10 X 10 or 20 X 20 pixels), we deal with each block as a two dimensional array of pixels (matrix) (Fig. 2). In order to illustrate the algorithm,
let us further assume that each block lies in the first quadrant of a Cartesian coordinate system where Y coordinate points up and X points to the right and the origin is located at the lower left corner of the block.

The algorithm replaces the pixels along a boundary within each block by a straight-line joining its two end pixels, which are located on the sides (boundaries) of the block. The algorithm deals with end pixels only instead of tracing pixels one by one. In other word, this algorithm marks the pixel as an end pixel (key point) if the location of the pixel falls within the block boundaries. The end pixels of a straight-line segment are pixels in the original boundary (Fig. 2). We calculate the slope and length of each straight-line segment directly from the pixels coordinates (i, j) in the block, by the following formulas:

\[
\text{Slope}(s) = \frac{i' - i}{j' - j} \quad (2)
\]

\[
\text{Length}(L) = \sqrt{(i' - i)^2 + (j' - j)^2} \quad (3)
\]

where \(i=0,1,2,...,n-1\) and \(j=0,1,2,3,...,n-1\)

We will call each straight-line (discrete segment) formed by two end-pixels primitive (P). The algorithm traverses the blocks that contain the boundary segments in a specific (clockwise or anticlockwise) direction until it reaches the first block where it started. Since all blocks are identical (same size), a lookup-table can be generated, which contains slope and length for all possible lines formed by end-pixels in the block. Using lookup-table will facilitate and fasten the process of representation and description. After having created the lookup-table, finding the slope and length for each segment is rather simple. We only need to identify the end pixels for each block.

**Shape description:**

Low computation complexity is a desirable characteristic of any shape descriptor. In our method a sequence of the lengths and the differences between the slopes of adjacent primitives can be used as a reliable descriptor of the boundary.

The difference in the slopes between any primitive \(P_{k+1}\) and the previous adjacent primitives \(P_k\) by can be calculated by the following formula

\[
D(S_{P_{k+1}}, S_{P_{k}}) = \frac{SP_{k+1} - SP_{k}}{1 + (SP_{k+1})(SP_{k})} \quad (4)
\]

The length (L) and the differences between the slopes of adjacent segments (D) are handled together to enrich shapes description, classification and querying capabilities. The parameter (D) describes how each segment of the boundary is connected to adjacent segment (Fig. 3). We apply the same process on the query
image and images in database (at the time of insertion into database). As might be expected, the accuracy of the resulting representation depends on the size of the blocks. The emphasis is not only on the accuracy of the representation, but also on efficiency (i.e. speed) of the operation. We illustrate our method by the following example.

Fig 2:  a) Dividing the original boundary into fixed size blocks, b) Replacing the boundary pixels by straight-lines

In Fig. 2 where the original boundary is divided into fixed size 10 X 10 blocks (matrices), the end pixels of P1 are (9,3) and (0,1), while for P2 they are (9,1) and (0,7). Based on equation (2) and (3) we find the slope (S) and the length (L) of primitives P1 and P2 respectively and the difference in the slope (D) between P1 and P2 can be derived from equation (4)

\[
\begin{align*}
S_{P1} &= -4.5 \\
L_{P1} &= 9.22 \\
S_{P2} &= 1.50 \\
L_{P2} &= 10.82 \\
D &= -1.04
\end{align*}
\]

The change in angle (\( \Delta \theta \)) of P2 with respect to P1 = \( \tan^{-1}(D) = -46.22^\circ \), the relation between P1 and P2 is illustrated in Fig. 3.

The natural characteristic of this shape descriptor makes itself invariant to translation. Furthermore, it can be made invariant to scale by a simple normalization, where the total length of all primitives that represent the shape divides the length of each primitive. Since the differences in the slopes between adjacent primitives
The change in angle ($\Delta \theta$) of P2 with respect to P1 remain unchanged no matter how you geometrically transform a shape. We treat this descriptor as a circular sequence of primitives. Circular shifting makes it robust to rotation. A shape in database is considered the best match to the query shape if they have the minimum difference between their primitives Sequences (descriptors). Primitives Merging. As stated earlier, the output of the representation process on an object boundary will be a sequence of discrete straight segments associated with their (L) and (S) parameters. For the shape in Fig. 2a the output of the representation process is shown in Fig. 4.

![Diagram](image)

**Fig. 4:** a) The original boundary, b) The boundary representation

The original boundary shape can be represented with possibly a large number of primitives or segments. Large number of primitives can cause the shape descriptor to be more complicated and the size of the index to increase unnecessarily. To avoid these problems, allowing adjacent primitives with similar slopes to be replaced by a single primitive can eliminate some primitives. For any two adjacent Primitives, if the difference between their slopes is zero or not more than a predetermined value (threshold), the primitives are considered similar and should be merged. This process will help to capture the essence of the boundary shape with the fewest possible segments and simplify the shape descriptor.

### 2. Conclusions
In the content based image retrieval, Shape and representation and description is essential. Low computation complexity is a desirable characteristic of any shape descriptor. In this paper we have presented a simple and efficient method for object shape description. Based on this method the object shape is represented and described as a sequence of discrete segments associated with their lengths and the differences in the slope between adjacent segments. The natural characteristics of this shape descriptor make it self invariant to translation. It can be made invariant to scale by a simple normalization. We treat this descriptor as a circular sequence of discrete segments. Circular shifting of the sequence makes it robust to rotation.

REFERENCES

1) ANILJ, K.J. and A. Vailaya, 1996. Image retrieval using color and shape. Pattern Recognition, 29: 8


3) Eakins, J.P. and M.E. Graham, 1999. Content-based image retrieval, a report to the JISC Technology Applications Programme. Institute for Image Data Research, University of Northumbria at Newcastle


