Journal of Computer Science 2 (7): 572-576, 2006
ISSN 1549-3636
© 2006 Science Publications

2D Object Description with Discrete Segments

Nbhan D. Salih, David Chek Ling Ngo and Hakim Mellah
Faculty of Information Science and Technology, Multimedia University (Melaka Campus)
Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia

Abstract: 2D object shape description has gained considerable attention in field of content-based image retrieval (CBIR) and in various applications that concerned with the shapes of objects. Object shape descriptors in the literature are commonly classified into two forms: contour-based and region-based. In this paper we present our 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.

Key words: CBIR, image retrieval, shape descriptor, shape representation and description

INTRODUCTION

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.

Basically, 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[1,2].

Eakins in[3] defined Content-based image retrieval CBIR as a *"technique for retrieving images on the basis of automatically-derived features such as colour, 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[4].

Since shapes bear semantic meanings, Shape is considered as the most important feature among all features that are used for querying the content of images[5,6]. 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 corrupted with noise, many systems are relatively successful in retrieving by color and texture, but perform poorly when searching on shape[7,8].

Shape representation and description generally looks for effective ways to capture the essence of 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[9]. 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[10] 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[11].

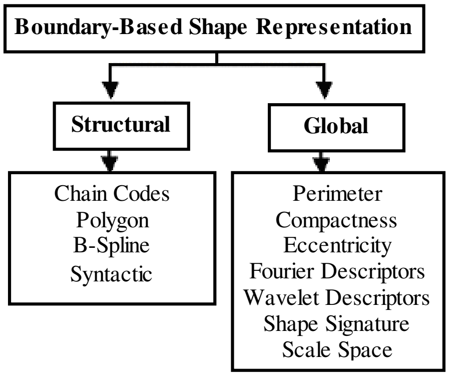

Fig. 1: Classification of boundary-based shape representation

There are many shape representation and description techniques in the literature. Zhang *et al.* in[8]

Corresponding Author: Nbhan D. Salih, Faculty of Information Science and Technology, Multimedia University (Melaka Campus), Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia

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 Fig. 1.

The most successful representatives for global boundary-based method is Fourier Description[4,8], Fourier Descriptor is obtained by applying Fourier transform on shape boundary. It was proposed by Persoon and Fu[13] in 1977. Many research have been done based on Fourier Descriptor[14-18]. However, Fourier Descriptor is sensitive to the starting point of the shape boundary, if the starting point changes the whole boundary sequence will change[15]. 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:

$$\mathbf{P} = \text{p}1, \text{p}2, \ldots, \text{p}n \qquad (1)$$

Where pi may be an element of a chain code, a side of a polygon, a spline, etc. pi may contain a number of attributed like length, average curvature, maximal curvature, orientation etc.,[8].

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[19]. 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 over come this problem the first difference of the chain code is used instead of the code itself[2,8].

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[11]. Split and merge[20,21] 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[22,11]. 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 codewords. 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 codewords[8,11].

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 strait-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:

$$Slope(\mathbf{S}) = \frac{i - i'}{j' - j} \qquad (2)$$

$$Length(\mathbf{L}) = \sqrt{(i' - i)^2 + (j' - j)^2} \qquad (3)$$

i = 0,1,2,….n-1 , j = 0,1,2,….n-1
Where n is the block size n X n

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 Pk+1 and the previous adjacent primitives Pk by can be calculated by the following formula:

$$\mathbf{D}(S_{P(k+1)}, S_{P(k)}) = \frac{S_{P(k+1)} - S_{P(k)}}{1 + (S_{P(k+1)})(S_{P(k)})} \qquad (4)$$

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.

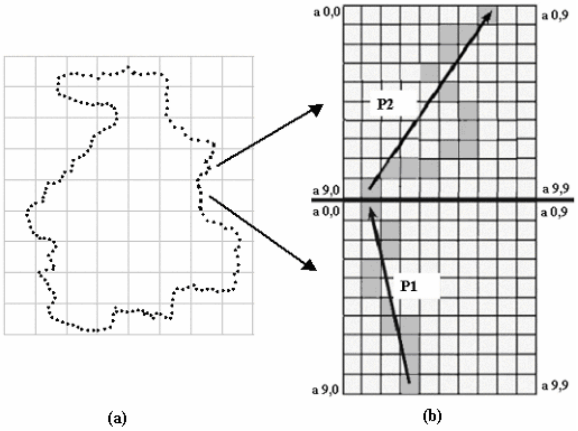

Fig 2: a) Dividing the original boundary into fixed size blocks, b) Replacing the boundary pixels by straight-lines

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)

Sp1= - 4.5	Lp1= 9.22
Sp2= 1.50	Lp2= 10.82
D = -1.04	

The change in angle ($\Delta\theta$) of P2 with respect to P1 = $\tan^{-1}$ (D) = - 46.22°, 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

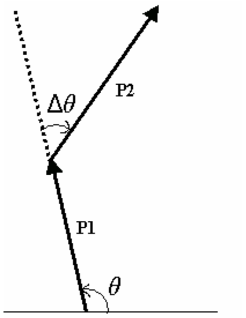

Fig. 3: The change in angle ($\Delta\theta$) of P2 with respect to P1

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.

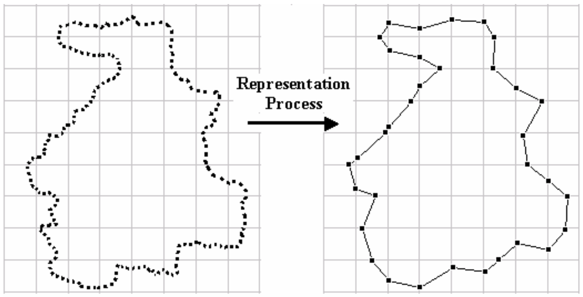

Fig. 4: a) The original boundary, b) The boundary representation

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.

CONCLUSION

Shape representation and description is essential in the content-based image retrieval. Low computation complexity is a desirable characteristic of any shape descriptor. In this paper, we have presented a simple and efficient method for 2D 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 characteristic of this Shape descriptor makes itself 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. Anil, K.J. and A. Vailaya, 1996. Image retrieval using color and shape. Pattern Recognition, 29: 8
2. Heesch, D., J. Hoare, M Gardoni, D. Gillies and S. Rüger, 2001. Content-based sketch retrieval and relevance feedback. Proc. SSGRR 2001
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
4. Rui, Y., T. Huang and S. Chang, 1999. Image retrieval: current techniques, promising directions and open issues. J. Visual Communication and Image Representation, 10: 4.
5. Saykol, E., U. Gudukbay and O. Ulusoy, 2005. A histogram-based approach for object-based query-by-shape-and-color in image and video databases. Image and Vision Computing, 23: 13
6. Dalong, L and S. Simske, 2002. Shape retrieval with flat contour segments. Technical Reports**,** Intelligent Enterprise Technologies Laboratory HP Laboratories, HPL-250-2002
7. Veltkamp, R.C. and M. Hagedoorn, 1999. State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht
8. Zhang, D. and G. Lu, 2004. Review of shape representation and description techniques. Pattern Recognition 37: 1-19
9. Kim, S., S. Kim, J. Kim and M. Kim, 2002. Usefulness of boundary sequences in computing shape features for arbitrary shaped regions. 16th Intl. Conf. Pattern Recognition (ICPR 2002), Canada, pp: 355-358

10. Schomaker, L., E. de Leau and L. Vuurpijl, 1999. Using pen-based outlines for object-based annotation and image-based queries. Proc. 3rd Intl. Conf. VISUAL'99, Amsterdam, pp: 585-592.
11. Loncaric, S., 1998. A survey of shape analysis techniques. Pattern Recognition, 31: 983-1001.
12. Zhang, D.S. and G. Lu., 2005. Study and Evaluation of Different Fourier Methods for Image Retrieval. International Journal of Computer Vision, 23(1):33-49
13. Persoon, E. and K.S. Fu, 1977. Shape discrimination using fourier descriptors. IEEE Trans. Systems, Man and Cybernetics, SMC-7:170-179.
14. Kauppinen, H., T. Seppanen and M. Pietikainen, 1995. An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. IEEE Trans. Pattern Analysis and Machine Intelligence, 17: 201-207
15. Yong, R., C.A. She and T.S. Huang, 1996. Modified fourier descriptors for shape representation-A practical approach. Proc. 1st Intl. Workshop on Image Databases and Multi Media Search, Amsterdam.
16. Rauber, T.W., 1994. Two-dimensional shape description. Technical Report: GR UNINOVA-RT-10-94, University Nova de Lisboa, Portugal.
17. Eichmann, G., M. Jankowski and R. Tolimieri, 1990. Shape representation by Gabor expansion. SPIE, Hybrid Image and Signal Processing II, 1297: 86-94
18. Rafiei, D. and A. Mendelzon, 2002. Efficient retrieval of similar shapes. The VLDB J., 11: 17-27.
19. Freeman, H., 1961. On encoding arbitrary geometric configurations. IRE Trans. Electronic Computers, 10: 260-268.
20. Ray, B. and K. Ray, 1995. A new split-and-merge technique for polygonal approximation of chain coded curves. PRL, 16: 161-169
21. Jain, R., R. Kasturi and B.G. Schunck, 1995. Machine Vision. McGraw Hill, pp: 188-210.
22. Witkin, A.P., 1983. Scale-space filtering. Proc.8th Intl. Joint Conf. Artificial Intelligence, pp: 1019-1022.