IMAGE RETRIEVAL USING CANONICAL CYCLIC STRING REPRESENTATION OF POLYGONS

Ömer Muhammet Soysal[1], *Bahadir Gunturk*[2], *Kenneth L. Matthews II*[3]
Louisiana State University,
[1]Department of Information Systems and Decision Sciences, osoysa1@lsu.edu
[2]Department of Electrical and Computer Engineering, bahadir@ece.lsu.edu
[3] Department of Physics & Astronomy, kipmatth@lsu.edu

ABSTRACT

In image retrieval applications one of the boundary-dependent approaches is matching contours with their polygonal representation. We introduce (1) a new polygonal shape representation, (2) an efficient algorithm to compute a unique representation of a polygon to handle orientation and (3) a matching method that is invariant to rigid and affine transformation. In the method, polygons are represented by a sequence of distance vectors ordered in a predefined cyclic way. Each vector is composed of two primitives which are radial distance from the centroid to a vertex and the following edge distance in a specified direction. Matching of polygons is achieved by bitwise comparison of their string code. The algorithm has a computational complexity of O(n log n), hence it has advantage for practical use.

Index Terms— Image coding, Image databases, Image matching, Image representations, Image shape analysis, String matching, Correlation.

1. INTRODUCTION

Image retrieval is an important task due to the increasing amount of image data employed in automated decision making processes. Shape representation as a way of abstraction is used to decrease the search space in image retrieval processes. Numerous models have been defined for shape representation. Each representation has its own primitives employed to find the best match. Shape primitives can be grouped based on utilizing radial distance from an origin, edge segments, and angles [1-8].

Representing shapes can be achieved using vertex-based or edge-based methods. The former has less computation time, while the latter is more accurate [2]. Among the edge-based methods, polygonal representation of shapes has been studied extensively [2]. Matching of polygons has been achieved by string representation using the three primitives mentioned above [5, 7]. These three primitives can be organized in different combinations to obtain a unique representation. Because computing a segment distance is less costly then computing an angle, radial and line segment distances are chosen in this study.

An orientation invariant representation has been obtained by shifting [7, 3, 11] or using max/min distance [8, 10]. We propose an algorithm to find a unique starting point. After representation is obtained, matching is achieved by comparison of corresponding elements of the representation.

In the matching process, error [3, 10], correlation [12], Hausdorf distance [8], normalized central moments [8], and edit distance of cyclic strings [7] have been employed as a similarity measure. In this study, we define *normalized correlation and error score* as our similarity measure.

The computational complexity for matching of two polygons with n vertices has been reported as either O(n^2) [13] or O(n^2 log n) [11, 14] to the best of our knowledge. Our proposed method has O(n log n) computational complexity for representation and Θ(n) for matching. Hence if shapes are stored with the proposed representation, matching can be done in Θ(n), which is the optimum, and online matching can be achieved in O(n log n) after obtaining vertices.

2. SHAPE REPRESENTATION

Definitions: Let $P = \langle V_1, V_2, ..., V_n \rangle$ denote a polygon of n vertices $V_i = (x_i, y_i)$ listed in counter-clockwise direction and let $c = (x_c, y_c)$ denote the centroid of P where $x_c = \sum_{i=1}^{n} x_i / n$, $y_c = \sum_{i=1}^{n} y_i / n$ and let $\delta_i = \langle \delta_{1i}, \delta_{2i} \rangle =$ be a distance vector where $\delta_{1i} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$ and $\delta_{2i} = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$. Let $P_C = \langle \delta_1, \delta_2, ..., \delta_n \rangle$ denote a cyclic canonical representation of P using distance vectors. (Note that, $\delta_{i \oplus n} = \delta_i$ where $1 \le i \oplus n = \text{mod}(i + n, n) \le$ n for $1 \le i \le n$.) Let $P_N = \langle \delta_a, \delta_{a\oplus 1}, ..., \delta_{a\oplus(n-1)} \rangle$ denote the normalized cyclic canonical representation of a polygon (NCCRP) where δ_a is a reference anchor computed by the algorithm NORMALIZE STRING given below.

Computation of max distance vector: Given two distance vectors δ_1 and δ_2, and a threshold τ, we say $\delta_1 > \delta_2$ if $\delta_{11} - \delta_{12} > \tau$ or if $|\delta_{11} - \delta_{12}| \le \tau$ and $\delta_{21} - \delta_{22} > \tau$.

Computation of max sequence: Given two sequences $T_1 = \langle \delta_1^1, \delta_2^1, ..., \delta_r^1 \rangle$ and $T_2 = \langle \delta_1^2, \delta_2^2, ..., \delta_r^2 \rangle$, we say $T_1 > T_2$ if $\delta_1^1 > \delta_1^2$ or if $\delta_j^1 = \delta_j^2$ and $\delta_{j+1}^1 > \delta_{j+1}^2$, $1 \le j < r$.

Computation of max segment:

Let $H = H_i = \{\delta_i : \delta_i = \max_{j=1,2,...,n} (\delta_j)\}$, for some $n > 0$, denote a *head* and T denote a *tail,* which is the subsequence between two successive heads H_i and $H_{i\oplus 1}$, and let $\hat{S}_i = \langle H_i, T_i \rangle$ denote a segment of P_C. We compute max of two segments $\hat{S}_1 = \langle H_1, T_1 \rangle$ and $\hat{S}_2 = \langle H_2, T_2 \rangle$ using

$$\arg max(\hat{S}_1, \hat{S}_2) = \begin{cases} \arg max(|T_1|, |T_2|) & \text{if } |T_1| \ne |T_2| \\ \arg max(T_1, T_2 |) & \text{if } |T_1| = |T_2| \end{cases} \quad (1)$$

Computation of parent segments: Let $\hat{S}^{(k)}$ denote the set of segments at iteration k and $I^{(k)}$ denote the starting index of segments to be computed at iteration $k + 1$.

$$\hat{S}^{(k)} = f(\hat{S}^{(0)}, I^{(k-1)}) = \langle \hat{S}_1^{(k)}, \hat{S}_2^{(k)}, ..., \hat{S}_{m_{k-1}}^{(k)} \rangle$$
$$= \langle \langle H_1^{(k)}, T_1^{(k)} \rangle, \langle H_2^{(k)}, T_2^{(k)} \rangle, ..., \langle H_{m_{k-1}}^{(k)}, T_{m_{k-1}}^{(k)} \rangle \rangle \quad (2.1)$$

where $\hat{S}^{(0)} = \langle \langle \delta_1 \rangle, \langle \delta_2 \rangle, ..., \langle \delta_n \rangle \rangle$, head $H_t^{(k)} = \delta_{I^{(k-1)}(t)}$ and tail $T_t^{(k)} = \langle \delta_{I^{(k-1)}(t)\oplus 1}, \delta_{I^{(k-1)}(t)\oplus 2}, ..., \delta_{I^{(k-1)}(t\oplus 1)-1} \rangle$ for $t = 1, 2, ..., m_{k-1}$, and

$$I^{(k)} = \{I^{(k-1)}(t) : \hat{S}_t^{(k)} = \max_{1 \le j \le m_{k-1}} (\hat{S}_j^{(k)})\} \quad (2.2)$$

Algorithm NORMALIZE STRING:

Strategy: Divide and conquer cyclically while constructing a tree in bottom-up form (An operation such as find, merge, compare are done cyclically.).
Goal: To find a unique ordering among the cyclic permutations of P_C.
Input: A cyclic string P_C.
Output: The normalized form P_N of string P_C.

1. [Initialization]: Set of initial segments: $\hat{S}^{(0)} = \langle \hat{S}_1^{(0)}, \hat{S}_2^{(0)}, ..., \hat{S}_n^{(0)} \rangle = \langle \langle \delta_1 \rangle, \langle \delta_2 \rangle, ..., \langle \delta_n \rangle \rangle$.
2. [Terminal layer construction]: Find all heads of $\hat{S}^{(0)}$ and store their indices in $I^{(0)} = \{i : \delta_i = \max_{1 \le j \le n} (\hat{S}_j^{(0)})\}$, $m_0 = |I^{(0)}|$. Let $I^{(0)}$ be a set of terminal nodes of the tree.
3. [Parent layer construction: $I^{(k)}$, $1 \le k \le K$]: Repeat until $m_{k-1} = 1$ or a symmetrical case ($m_{k-1} = m_k$) will be decided
   a. Compute the set of parent segments $\hat{S}^{(k)}$ from child segments $\hat{S}^{(k-1)}$ by (2.1).
   b. Find the max segments $\hat{S}_{t'}^{(k)} \in \hat{S}^{(k)}$, $1 \le t' \le m_{k-1}$, and store their index in $I^{(k)}$ by (2.2) using the following operations
      i. First compare the segments in length,
      ii. If more than one max segments exists, then lexicographically compare the segments having max length.
4. $a = I^K(1)$, $P_N = \langle \delta_a, \delta_{a\oplus 1}, ..., \delta_{a\oplus(n-1)} \rangle$;

For example, let $P_C = \langle (3,1), (2,1), (5,1), (3,1), (4,1), (5,1) \rangle$ then P_N will be computed as $P_N = \langle (5, 1), (3, 1), (4, 1), (5, 1), (3, 1), (2, 1) \rangle$ where segments are $\hat{S}_1 = \langle (5, 1), (3, 1), (4, 1) \rangle$ and $\hat{S}_2 = \langle (5, 1), (3, 1), (2, 1) \rangle$ with $H = \langle (5, 1) \rangle$ and $T_1 = \langle (3, 1), (4, 1) \rangle$ and $T_2 = \langle (3, 1), (4, 1) \rangle$, and $\delta_a = \delta_6$. A symmetrical example is $P_C = \langle (5, 1), (3, 1), (4, 1), (5, 1), (3, 1), (4, 1) \rangle$, so δ_a will be any of head.

Invariance property of the representation: NCCRP is invariant under similarity transformation. Affine invariance can be obtained by first fitting an ellipse to the polygon and normalizing it by a unit circle-mapping, and then computing NCCRP from the normalized version of the polygon. Figure 1 shows such normalization.

Complexity of the representation: NCCRP can be obtained in O(n log n) computational complexity from P_C. Informally, at the worse case there will be even number of heads at the beginning and their count will be halved at a successive iteration. Hence computation complexity will be of O(n log n).

Similarity properties: The similarity score satisfies i) reflexivity: $S(A, A) = 1$ ii) symmetry: $S(A, B) = S(B, A)$.

3. MATCHING AND ALIGNMENT

Computation of similarity score: Given two strings of query $Q = \langle q_1, ..., q_n \rangle$ and reference $R = \langle r_1, ..., r_n \rangle$ in P_N form, where q_i and r_i are distance vectors listed in the same specified direction say clock-wise. Let $u_i = q_i / \|q_i\|$ and $v_i = r_i / \|r_i\|$ be unit vectors for the distance vectors q_i and r_i, respectively. We define the *normalized correlation and error ratio* (*NCER*) as

$$NCER(Q, R) = \langle \phi_1, \phi_2, ..., \phi_n \rangle \quad (3)$$

where $\phi_i = u_i \cdot v_i \left(1 - \frac{\left| \|q_i\| - \|r_i\| \right|}{\|q_i\| + \|r_i\|} \right)$. We define the *normalized correlation and error score* (*NCES*) as

$$NCES(Q,R) = 1 - [\mathrm{stdev}(NCER(Q,R))/\mathrm{mean}(NCER(Q,R))] \quad (4)$$

Robustness of the matching: NCCRP is robust the minor deviations in vertex locations. A threshold can be specified for setting robustness to small deviations. If there are some anchor candidate vectors which are close to each other within a threshold, then P_N must be computed for each candidate anchor. Its accuracy is dependent on how well vertices are located. Test results show that even if vertices are located with some small errors, similarity is not affected too much.

Affine transformation alignment: Query and reference polygons can be aligned by means of any three points located at the same index of their NCCRP. Accuracy of locating salient vertex points plays an important role; expectedly, it affects the alignment accuracy.

4. RESULTS

Experimental Setup: We photographed physical shapes built from plastic bricks. Each image's size was 640x480 in pixel. Figure 2 shows the polygonal shapes generated from five of sample images. Rigid transformed versions of the images are obtained by taking images either by moving the camera to a different view or by rotating the object or adjusting the zoom level of the camera. Extracting vertices were done manually. (Methods for automated vertex extraction is not the concern of this study.) For additional data, a collection of 5000 synthetic polygons were randomly created. Out of these polygons, extra 5000 from both rigid and affine transformations were randomly created. Several query shapes were tested to retrieve them from the collection. Shapes were stored in a flat file with their vertices. The average elapsed time to find similar polygons among 10000 polygons including only their rigid transformed versions was 27 sec for polygons having 10 vertices, and 76 sec for 100 vertices. It took 118 sec to find similar polygons among 15000 polygons including rigid and affine transformed versions. In testing we used a 1.5 GHz PC.

Alignment of rigid body (similarity) transformed images: Image D1 was aligned with its rigid-transformed version Image D2 as shown in Figure 3. The transformation parameters computed as sx = 1.8687 and sy = 1.8361 for scale, shx = -0.0448 and shy = -0.0111 for shear, and tx = -8.3230 and ty = -7.2267 for translation. The small discrepancy from rigid body structure might be either due to shearing while moving camera or due to errors in determining vertices from the images.

Retrieval of similar images: Images A and B, and Images D1 and D2 were identified as similar. Table 1 summarize similarity measure results. While *NCER* for Images A and B is 0.92, *NCES* is 0.61. This shows Image A and B are similar but not so identical as compared to the perfect (100%) similarity score. Table 1 also indicates a large similarity score in the case of close similarity when comparing Image D1 and to its almost identical rigid transformed version Image D2. Even images are affine transformed, similar images within the database can be found by the approach proposed as seen in Table 1 for Images D1 and D2.

4. CONCLUSION

In this study a new polygonal shape representation, which is unique for a polygon, is introduced. A matching of polygonal shapes can be achieved using this representation in Θ(n) time. The algorithm to compute the representation has a computational complexity of $O(n \log n)$, hence it may be sufficiently fast for online image retrieval. It is robust and invariant under similarity transformation and it can be easily extended to handle affine transformations.

Acknowledgement: We like to thank Dr. Sukhamay Kundu for his suggestions in the development of canonical representation using the notion of longest lexicographic ordering.

5. REFERENCES

[1] C.C. Chang, S.M. Hwang, and D.J. Buehrer, "A Shape Recognition Scheme Based On Relative Distances Of Feature Points From The Centroid", *Pattern Recognition*, 24 (11): 1053-1063, 1991.

[2] J.M. Chen and J.A. Ventura, "Optimization models for shape matching of nonconvex polygons", *Pattern Recognition*, 28(6), 863-877, 1995.

[3] L.K. Huang, M.J. Wang, "Efficient shape matching through model-based shape recognition", *Pattern Recognition*, 29(2), 207-215, 1996.

[4] Z. Wang, Z. Chi, and D. Feng, "Shape based leaf image retrieval", *IEE Proc.-Vision Image And Signal Processing*, 150(1), 34-43, 2003.

[5] W. Wu, "Two-dimensional object recognition through string matching", *Imaging Science Journal*, 49(4), 213-221, 2001.

[6] J.H. Chuang, "A potential-based approach for shape matching and recognition", *Pattern Recognition*, 29(3), 463-470, 1996.

[7] M. Maes, "Polygonal Shape Recognition Using String-Matching Techniques", *Pattern Recognition*, 24(5), 433-440, 1991.

[8] H. Nishida, "Matching And Recognition Of Deformed Closed Contours Based On Structural Transformation Models",*Pattern Recognition*, 31(10), 1557-1571, 1998.

[8] S. Minsoo and K. Hwanil, "New Measures Of Similarity Between Two Contours Based On Optimal Bivariate Transforms", *CVIU*, 26(2), 168-182, 1984.

[10] T. Bernier and J.A. Landry, “A New Method For Representing And Matching Shapes Of Natural Objects”, *Pattern* Recognition*,* 36(8), 1711-1723, 2003.

[11] E.M. Arkin, L.P. Chew, D.P. Huttenlocher, et al., “An Efficiently Computable Metric For Comparing Polygonal Shapes”, *IEEE Trans. PAMI*, 13(3), 209-216, 1991.

[12] L. Vangool, T. Moons, D. Ungureanu and A. Oosterlinck, “The Characterization and Detection of Skewed Symmetry”, *CVIU*, 61(1), 138-150, 1995.

[13] I. Schreiber and M. Ben-Bassat, “Polygonal Object Recognition”, *IEEE Proc. 10th Int. Con. on Pattern Recognition*, 1, 852 – 859, 1990.

[14] M. Maes, “On A Cyclic String-To-String Correction Problem”, *Information Processing Letters*, 35(2), 73-78, 1990.

Similarity	A, B	C, B	A, C	D1, D2
Mean(NCR)	92.07%	64.61%	67.64%	97.71%
Stdev(NCR)	0.1319	0.2296	0.2159	0.0141
NCES	60.55%	25.84%	31.31%	87.83%

Table.1. Similarity scores

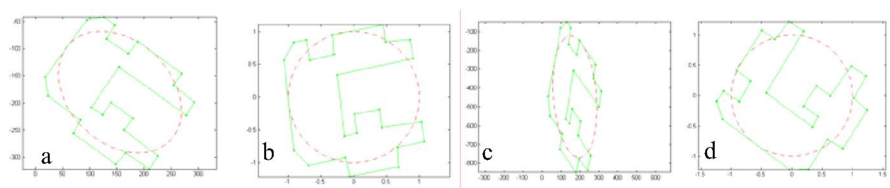

Figure 1. Ellipse fitting and normalization: a) Original image; b) Original image after normalization; c) Affine transformed image; d) Affine transformed image after normalization.

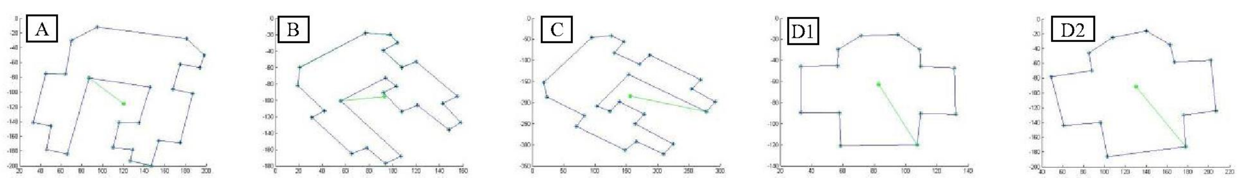

Figure 2. Sample images used in experiments. The line from the centroid shows the anchor in the clock-wise direction. (Images are scaled to fit)

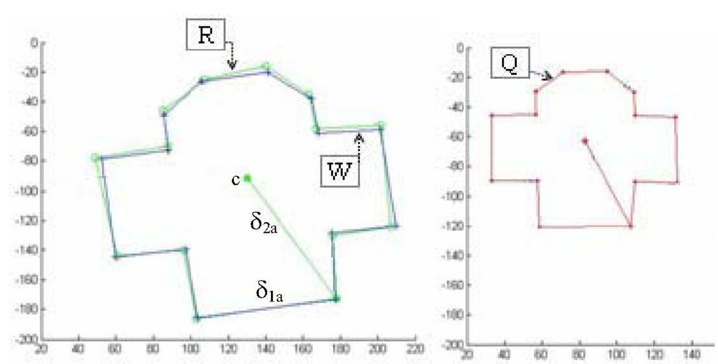

Figure 3. Rigid body alignment, *Q*: query, *R*: reference, and *W*: warped from *Q*